

## Distributed computer system

**Publication number:** DE19513922

**Publication date:** 1995-10-19

**Inventor:** SAITO MASAHIKO (JP); YOKOYAMA TAKANORI (JP);  
SHIMADA MASARU (JP); TSUNEDOMI KUNIIHIKO (JP);  
NAKAMURA TOMOAKI (JP)

**Applicant:** HITACHI LTD (JP)

**Classification:**



- international: **G06F15/16; G06F9/46; G06F9/48; G06F9/50;  
G06F15/177; G06F15/16; G06F9/46; (IPC1-7):  
G06F15/16; G06F9/46**

- european: **G06F9/46A4L; G06F9/46A4M**

**Application number:** DE19951013922 19950412

**Priority number(s):** JP19940075610 19940414

**Also published as:**

 US5845116 (A1)  
 JP7282013 (A)

**Report a data error here**

### Abstract of DE19513922

The computer system has a number of computers (100,...103), for common execution of several programmes, with a priority assignment device (144,145; 146,147) providing a priority value for execution of a programme in dependence on the function provision and/or loading of each computer and on the urgency value assigned to the programme. Pref. each computer has a priority value converter (144,145) for converting the programme urgency value into a corresponding priority value, dependent on the function provision and loading of the computer and a priority altering device (146,147) for altering the priority of the programme being executed by the computer in dependence on the priority value from the priority converter.

---

Data supplied from the **esp@cenet** database - Worldwide



①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ **Offenlegungsschrift**  
⑩ **DE 195 13 922 A 1**

⑤1 Int. Cl.<sup>6</sup>:  
**G 06 F 15/16**  
G 06 F 9/46

②1 Aktenzeichen: 195 13 922.4  
②2 Anmeldetag: 12. 4. 95  
④3 Offenlegungstag: 19. 10. 95

DE 195 13 922 A 1

③0 Unionspriorität: ③2 ③3 ③1  
14.04.94 JP 6-075610

⑦1 Anmelder:  
Hitachi, Ltd., Tokio/Tokyo, JP

⑦4 Vertreter:  
Strehl, Schübel-Hopf, Groening & Partner, 80538  
München

⑦2 Erfinder:  
Saito, Masahiko, Hitachi, JP; Yokoyama, Takanori,  
Hitachi, JP; Shimada, Masaru, Hitachi, JP;  
Tsunedomi, Kunihiko, Hitachi, JP; Nakamura,  
Tomoaki, Hitachinaka, JP

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤4 Verteiltes Computersystem

⑤7 Es wird ein verteiltes Computersystem mit mehreren Computern angegeben, die sich hinsichtlich des Funktionsvermögens, der Auslastung und des Typs voneinander unterscheiden, und das örtliche Prioritätsschemata, wie sie in den jeweiligen Computern vorliegen, einheitlich dadurch verwaltet, das entweder ein Konzept von "Dringlichkeitswerten" oder ein solches von "Zeitgrenzen" verwendet wird. Jeder der Computer verfügt über einen Prioritätswert-Umsetzablauf zum Ausführen einer Umsetzung zwischen einem Dringlichkeitswert und einem Prioritätswert für die Verarbeitung abhängig vom Funktionsvermögen und der Auslastung des Computers, und über einen Prioritätswert-Änderungsablauf zum Ändern des Prioritätswerts eines Programms, das die Verarbeitung ausführt, abhängig vom Prioritätswert, wie er vom Prioritätswert-Umsetzablauf angegeben wird.

DE 195 13 922 A 1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

BUNDESDRUCKEREI 08. 95 508 042/690

78,



Die Erfindung betrifft ein verteiltes Computersystem mit mehreren zu einem Netzwerk verschalteten Computern, um gemeinsam mehrere Programme abzuarbeiten, und insbesondere betrifft die Erfindung die Konfiguration und das Betriebsverfahren eines derartigen verteilten Computersystems, bei dem Prioritätsmaßnahmen verwendet werden, um Echtzeiteigenschaften zu realisieren.

Um bei einem herkömmlichen Computersystem, das nur aus einem einzelnen Computer besteht (wozu auch Computer mit mehreren Prozessoren gehören) Echtzeiteigenschaften zu schaffen, wurde eine Vorgehensweise verwendet, gemäß der die Reihenfolge der abzuarbeitenden Programme auf Grundlage einer Prioritätsreihenfolge der Programme gesteuert wird. Ein Beispiel für eine derartige Steuerung der Verarbeitungsreihenfolge unter Verwendung einer Prioritätsreihenfolge ist in "Operating Systems — Design and Implementation" von A. S. Tanenbaum, Prentice-Hall, Inc. S. 82—84 (1987) beschrieben.

Auch beschreibt z. B. die Schrift JP-A-2-113362 ein Steuerschema, bei dem ein Prioritätswert zu jeder Meldung hinzugefügt wird, wenn eine solche zwischen Verarbeitungsknoten innerhalb eines verteilten Rechnersystems übertragen wird, damit diese Meldungen in der Prioritätsreihenfolge verarbeitet werden, ausgehend mit der Meldung höchster Priorität (dieses System wird nachfolgend als "Stand der Technik 1" bezeichnet).

Ein anderes herkömmliches Schema ist in der Schrift JP-A-5-35701 beschrieben. Dieses Schema spezifiziert die tolerierbare Zeit für jede anhängige Verarbeitung, wenn eine Verarbeitungsanforderung in einem verteilten Computersystem mit mehreren Prozessoren ausgegeben wird, es untersucht die geplante Verarbeitungsanhängigkeitszeit jedes Prozessors, und es ordnet die Verarbeitungsanforderung einem Prozessor zu, dessen geplante Verarbeitungsanhängigkeitszeit kürzer als die tolerierbare Verarbeitungsanhängigkeitszeit ist. Wenn kein Prozessor der Bedingung der tolerierbaren Verarbeitungsanhängigkeitszeit genügt, wird die Verarbeitungsanforderung demjenigen Prozessor zugeordnet, der die kürzeste geplante Verarbeitungsanhängigkeitszeit aufweist. Verarbeitungsanforderungen, die jeweiligen Prozessoren zugeordnet sind, werden grundsätzlich von der ältesten Anforderung her abgearbeitet (dieses System wird nachfolgend als "Stand der Technik 2" bezeichnet).

Ferner ist es z. B. in der Schrift JP-A-5-35701 beschrieben, daß dann, wenn Verarbeitungsanforderungen ausgegeben werden, zu jeder solchen Anforderung eine tolerierbare Verarbeitungsanhängigkeitszeit wie auch ein Prioritätswert hinzugefügt werden, ähnlich wie beim Stand der Technik 1, damit die Verarbeitungsanforderungen nach der Zuordnung zu einem Prozessor vom jeweilig zugeordneten Prozessor in der Reihenfolge der Prioritätswerte abgearbeitet werden (dieses Verfahren wird nachfolgend als "Stand der Technik 3" bezeichnet).

Bei einem verteilten Computersystem, in dem mehrere Computer durch jeweilige ihnen zugeordnete Betriebssysteme verwaltet werden, wird die Priorität der Programme in jedem der Computer gesondert verwaltet. Daher kann die Priorität von Programmen, wie sie von den jeweiligen Computern auszuführen sind, nicht gleichmäßig im gesamten verteilten Computersystem verwaltet werden. Insbesondere in einem verteilten Computersystem mit Computern, die sich hinsichtlich

der Funktionsfähigkeit und der Auslastung unterscheiden, kann sich die Dringlichkeit, wie sie durch die Priorität in einem Computer repräsentiert wird, von der Dringlichkeit unterscheiden, wie sie von der Priorität in einem anderen Computer repräsentiert wird. Genauer gesagt, kann dann, wenn ein Computer mit hoher Leistungsfähigkeit mit einem solchen mit geringer Leistungsfähigkeit verglichen wird, der erstere eine viel schnellere Verarbeitung als der letztere ausführen, und zwar selbst dann, wenn sie dieselbe Verarbeitung ausführen, die einen auf null gesetzten Prioritätswert hat. Auf ähnliche Weise unterscheiden sich bei einem stark ausgelasteten Computer und einem wenig ausgelasteten Computer die Bedeutungen hinsichtlich ihrer jeweiligen Prioritätswerte voneinander.

Darüber hinaus können sich bei einem verteilten Computersystem mit mehreren verschiedenen Computern die Prioritätsschemata selbst von einem Computer zum nächsten verschieden sein. Z.B. ist ein Computer mit einem Prioritätswertebereich von 0 bis 255 nicht mit einem Computer mit einem Prioritätswertebereich von 0 bis 127 hinsichtlich der Werte kompatibel, die den Prioritätsrängen zugeordnet werden. Auf ähnliche Weise kann in einem Computer, der den Prioritätswert 0 als den höchsten Rang ansieht, nicht derselbe Wert zugeordnet werden wie in einem Computer, der den Prioritätswert 255 als höchsten Rang ansieht.

Der Stand der Technik 1 überträgt einfach eine Meldung zusammen mit einem Prioritätswert, ohne die Unterschiede hinsichtlich des Funktionsvermögens, der Auslastung und des Typs verschiedener Computer in einem verteilten Computersystem zu berücksichtigen. Wenn eine Meldung an einen Computer mit geringem Funktionsvermögen oder an einen stark ausgelasteten Computer geliefert wird, ist es wahrscheinlich, daß die Verarbeitung nicht mit gewünschter Geschwindigkeit ausgeführt wird, was möglicherweise dazu führt, daß die Echtzeiteigenschaften nicht aufrechterhalten werden können.

Ferner kann beim Stand der Technik 1, dann, wenn mehrere Computer dazu aufgefordert werden, Programme auszuführen, ein umgekehrter Effekt auftreten, d. h., daß eine Anforderung mit niedrigerem Prioritätsrang schneller abgeschlossen wird als eine Anforderung mit höherem Prioritätsrang. Es sei z. B. angenommen, daß eine Verarbeitung A mit dem Prioritätswert 5 an einen Computer A gegeben wird, während eine Verarbeitung B mit dem Prioritätswert 1 an einen Computer B gegeben wird (es sei auch angenommen, daß ein höherer Prioritätsrang durch einen kleineren Prioritätswert repräsentiert ist). In diesem Fall sollte die Verarbeitung B schneller abgeschlossen werden als die Verarbeitung A, um dem ursprünglichen Zweck zu genügen. Wenn jedoch das Funktionsvermögen des Computers B schlechter als das des Computers A ist oder wenn der Computer B stärker ausgelastet ist als der Computer A, ist es möglich, daß die Verarbeitung A mit geringerer Priorität schneller abgeschlossen wird als die Verarbeitung B mit höherer Priorität.

Ferner führt beim Stand der Technik 1 ein Computer, der eine große Anzahl von Anforderungen mit höheren Prioritätsrängen empfangen hat, zu Schwierigkeiten beim Sicherstellen von Echtzeiteigenschaften, da keine Situation berücksichtigt ist, bei der Verarbeitungsanforderungen auf einen einzelnen Computer konzentriert sind, insbesondere eine Situation, bei der Verarbeitungsanforderungen mit höheren Prioritätsrängen auf einen einzelnen Computer konzentriert sind.



Der Stand der Technik 2 kann dazu verwendet werden, Echtzeiteigenschaften in gewissem Ausmaß hinsichtlich einzelner Verarbeitungsanforderungen zu gewährleisten, vorausgesetzt, daß die geplante Verarbeitungsanhängigkeitszeit für die jeweiligen Prozessoren korrekt berechnet werden kann. Jedoch kann dieses System keine Zeitgrenzebedingung hinsichtlich einer Verarbeitungsanforderung mit kurzer tolerierbarer Verarbeitungsanhängigkeitszeit, die neu ausgegeben wird, nachdem eine bestimmte Anzahl von Verarbeitungsanforderungen bereits jeweiligen Prozessoren zugeordnet sind, erfüllen. Z.B. können Situationen auftreten, bei denen ein Prozessor dazu in der Lage war, neu ausgegebene Verarbeitungsanforderungen vor Verarbeitungsanforderungen mit langen tolerierbaren Verarbeitungsanhängigkeitszeiten, wie sie bereits Prozessoren zugeordnet waren, zu verarbeiten. Dabei könnte die Zeitgrenzebedingung für alle Verarbeitungsanforderungen erfüllt sein, jedoch ist beim System gemäß dem Stand der Technik 2, das Verarbeitungsanforderungen der Reihe nach zuordnet, die Zeitgrenzebedingung gelegentlich für neu ausgegebene Verarbeitungsanforderungen nicht erfüllt. Anders gesagt, kann der Stand der Technik 2 eine hochdringliche Verarbeitungsanforderung, die bei relativ stark ausgelasteter Situation ausgegeben wird, nicht unmittelbar verarbeiten. Demgemäß hat der Stand der Technik 2 ein Problem dahingehend, daß strenge Echtzeiteigenschaften nicht gewährleistet sind.

Der Stand der Technik 3 ist andererseits ein System, das folgendes umfaßt: (1) Auswählen eines Prozessors, der einer tolerierbaren Verarbeitungsanhängigkeitszeit genügt, wenn einem Prozessor eine Verarbeitungsanforderung zugeordnet wird; und (2) Ausführen von Verarbeitungsanforderungen in der Reihenfolge der Prioritätswerte, wie sie zu den jeweiligen Anforderungen in jedem Prozessor hinzugefügt sind. Beim Stand der Technik 3 kann der Nachteil des Stands der Technik 2 in gewissem Ausmaß dadurch überwunden werden, daß einer neu ausgegebenen Verarbeitungsanforderung mit kurzer tolerierbarer Verarbeitungsanhängigkeitszeit ein hoher Prioritätsrang zugeordnet wird. Da dieses System jedoch vorzugsweise neu zugeordnete Verarbeitungsanforderungen mit hohem Prioritätsrang verarbeitet, kann es unter Umständen der tolerierbaren Verarbeitungsanhängigkeitszeit einer bereits zugeordneten Verarbeitungsanforderung mit niedrigem Prioritätsrang nicht genügen, wobei der niedrige Prioritätsrang deswegen festgelegt wurde, da angenommen wurde, die tolerierbare Verarbeitungsanhängigkeitszeit würde erfüllt. Es sei z. B. eine Situation angenommen, bei der eine Verarbeitungsanforderung A mit einer auf eine Minute gesetzten tolerierbaren Verarbeitungsanhängigkeitszeit und einem Prioritätswert 5 einem Prozessor A zugeordnet wurde, wobei dies erfolgte, da angenommen wurde, daß die tolerierbare Verarbeitungsanhängigkeitszeit erfüllt wurde, und daß eine Verarbeitungsanforderung B mit einer auf 30 Sekunden festgesetzten tolerierbaren Verarbeitungsanhängigkeitszeit und mit dem Prioritätswert 1 dem Prozessor A neu zugeordnet wird. In diesem Fall wird die Verarbeitungsanforderung B vor der Verarbeitungsanforderung A verarbeitet. Diese bevorzugte Ausführung kann dazu führen, daß die tolerierbare Verarbeitungsanhängigkeitszeit für die Verarbeitungsanforderung A, die bereits demselben Prozessor zugeordnet war, nicht erfüllt werden kann, obwohl auch diese tolerierbare Verarbeitungsanhängigkeitszeit sicher erfüllt werden sollte.

Auch kann beim Stand der Technik 3 dann, wenn

verschiedene Verarbeitungsanforderungen mit derselben tolerierbaren Verarbeitungsanhängigkeitszeit jedoch verschiedenen Prioritätswerten mehreren Prozessoren zugeordnet werden, ein umgekehrter Effekt, ähnlich wie beim Stand der Technik 1, wegen verschiedener Auslastungen und Funktionsfähigkeiten der zugeordneten Prozessoren auftreten, daß also eine Verarbeitungsanforderung mit niedrigerem Prioritätsrang schneller abgeschlossen wird als eine Verarbeitungsanforderung mit höherem Prioritätsrang.

Der Erfindung liegt die Aufgabe zugrunde, ein verteiltes Computersystem zu schaffen, in dem Echtheitseigenschaften strenger eingehalten werden als bei bisherigen derartigen Systemen.

Es ist eine andere Aufgabe der Erfindung, ein verteiltes Computersystem zu schaffen, das Computer enthält, die sich möglicherweise hinsichtlich des Funktionsvermögens, der Auslastung und des Typs unterscheiden, und das dazu in der Lage ist, Echtzeiteigenschaften strenger dadurch einzuhalten, daß örtliche Prioritätsschemata, wie sie in den jeweiligen Computern vorliegen, gleichmäßig verwaltet werden.

Die Erfindung verwendet eines der folgenden zwei Konzepte, um die örtlichen Prioritätsschemata gleichmäßig zu verwalten, die in den verschiedenen Computern vorliegen, die zum verteilten Computersystem gehören:

- (1) Verarbeitungsdringlichkeit: dies ist ein gemeinsamer Relativwert, wie er im gesamten System dazu verwendet wird, anzuzeigen, wie schnell eine Verarbeitung abzuschließen ist; und
- (2) Verarbeitungszeitgrenze: dies ist ein absoluter Wert, der anzeigt, wie schnell eine Verarbeitung vorzugsweise abzuschließen ist.

Nachfolgend werden Einrichtungen für die gleichmäßige Verwaltung der örtlichen Prioritätsschemata in den jeweiligen Computern in einem verteilten Computersystem, wie bei der Erfindung verwendet, getrennt für den Fall beschrieben, daß die Verarbeitungsdringlichkeit verwendet wird, und für den Fall, daß die Verarbeitungszeitgrenze verwendet wird.

#### (A) Verwaltungseinrichtungen mit Verarbeitungsdringlichkeit

Um die vorstehend genannten Aufgaben zu lösen, ist die Erfindung gemäß einer Erscheinungsform durch eine Prioritätswert-Bestimmungseinrichtung zum Bestimmen des Prioritätswerts gekennzeichnet, wie er zum Ausführen eines Programms geeignet ist, und zwar unter Berücksichtigung der Funktionsfähigkeit und der Auslastung jedes Computers und/oder des Dringlichkeitswerts für die vom Computer ausgeführte Verarbeitung.

Insbesondere enthält die Prioritätswert-Bestimmungseinrichtung die folgenden fünf Einrichtungen als Komponenten, die zusammenwirkend miteinander arbeiten, um die Verwaltung eines verteilten Computersystems in solcher Weise auszuführen, daß die vorstehend genannten Aufgaben unter Verwendung der Verarbeitungsdringlichkeit gelöst werden:

- (1) Prioritätswert-Umsetzeinrichtung zum Ausführen einer Umsetzung zwischen einem Dringlichkeitswert und einem zugehörigen Prioritätswert für die Verarbeitung abhängig vom Funktionsver-



mögen und der Auslastung eines Computers;

(2) Prioritätswert-Änderungseinrichtung zum Ändern des Prioritätswerts eines Programms, wodurch die Verarbeitung mit einem Prioritätswert ausgeführt wird, wie sie von der Prioritätswert- 5 Umsetzeinrichtung angegeben wird;

(3) Computerauswahleinrichtung zum Bestimmen eines Computers, der zum Ausführen des Programms geeignet ist, und zwar unter Berücksichtigung des Funktionsvermögens und der Auslastung 10 jedes Computers und/oder des Dringlichkeitswerts der von einem Programm ausgeführten Verarbeitung;

(4) Programmauswahleinrichtung, die arbeitet, wenn mehrere Programme, die denselben Vorgang ausführen, vorliegen und Verarbeitung für eines dieser Programme angefordert wird, um ein Programm mit optimalen Prioritätswert aus den mehreren Programmen auszuwählen, die denselben Vorgang ausführen, und zwar unter Berücksichtigung des Funktionsvermögens und der Auslastung 20 jedes Computers und/oder des Dringlichkeitswerts der Verarbeitung; und

(5) Verarbeitungsverlegungseinrichtung zum Verlegen einer neu angeforderten Verarbeitung eines Programms, wie von der Programmauswahleinrichtung angezeigt. 25

#### (B) Verwaltungseinrichtung für Verarbeitungszeitgrenze 30

Um die obige Aufgabe zu lösen, ist die Erfindung gemäß einer anderen Erscheinungsform durch eine Prioritätswert-Bestimmungseinrichtung zum Bestimmen des Prioritätswerts, wie er zum Ausführen eines Programms geeignet ist, gekennzeichnet, wobei das Funktionsvermögen und die Auslastung jedes Computers und/oder die Zeitgrenze und die Verarbeitungs- 35 menge der von einem Programm ausgeführten Verarbeitung berücksichtigt werden.

Speziell verfügt die Prioritätswert-Bestimmungseinrichtung über die folgenden acht Einrichtungen als Komponenten, die zusammenwirkend miteinander so arbeiten, daß sie die Verwaltung eines verteilten Computersystems dahingehend ermöglichen, daß die oben genannten Aufgaben unter Verwendung der Verarbeitungszeitgrenze gelöst werden. Es ist jedoch zu beachten, daß nicht alle folgenden acht Einrichtungen gleichzeitig vorhanden sein müssen: 40

(1) Prioritätswert-Umsetzeinrichtung zum Ausführen einer Umsetzung zwischen einem Wertesatz für eine Grenze und eine Verarbeitungsmenge für die Verarbeitung und einem Prioritätswert abhängig vom Funktionsvermögen und der Auslastung eines Computers; 50

(2) Prioritätswert-Änderungseinrichtung zum Ändern des Prioritätswerts eines Programms, das von einem Computer ausgeführt wird, abhängig vom Prioritätswert, wie er von der Prioritätswert-Umsetzeinrichtung angegeben wird; 55

(3) Verbindungszeit-Vorhersageeinrichtung zum Vorhersagen der Verbindungszeit mit einem anderen Computer;

(4) Computerauswahleinrichtung zum Bestimmen eines Computers, wie er zum Ausführen eines Programms geeignet ist, wobei das Funktionsvermögen und die Auslastung jedes Computers und/oder 65

ein Wertesatz einer Zeitgrenze und einer Verarbeitungsmenge für eine Verarbeitung, wie sie von einem Programm ausgeführt wird, berücksichtigt werden;

(5) Programmauswahleinrichtung, die dann arbeitet, wenn mehrere Programme, die denselben Vorgang ausführen, vorliegen und eine Verarbeitung durch eines dieser Programme angefordert wird, um das Programm mit optimalem Prioritätswert aus den mehreren Programmen auszuwählen, die denselben Vorgang ausführen, wobei das Funktionsvermögen und die Auslastung jedes Computers und/oder ein Wertesatz aus einer Zeitgrenze und einer Verarbeitungsmenge für die Verarbeitung berücksichtigt werden; 10

(6) Verarbeitungsverlegereinrichtung zum Verlegen einer neu angeforderten Verarbeitung für ein Programm, wie durch die Programmauswahleinrichtung angezeigt; 15

(7) Verarbeitungszeit-Vorhersageeinrichtung zum Vorhersagen einer Verarbeitungszeit, wie sie für ein Programm erforderlich ist, um die gerade ausgeführte Verarbeitung abzuschließen, und zwar abhängig vom Funktionsvermögen, der Auslastung, der Verarbeitungsmenge des Computers; und 20

(8) Zeitgrenze-Einstelleinrichtung, die dann arbeitet, wenn die vorhergesagte Verarbeitungszeit nicht der für die Verarbeitung auferlegten Zeitgrenze genügt, um eine Zeitgrenze neu einzustellen, ausgehend von der Zeit, die ab der Anforderung der Verarbeitung verstrichen ist, und ausgehend von der auferlegten Zeitgrenze. 25

Der Vorgang, wie er von der Einrichtung zum Überwinden der oben genannten Schwierigkeiten ausgeführt wird, wird getrennt für den Fall, daß die Verarbeitungsdringlichkeit verwendet wird, und für den Fall, daß die Verarbeitungszeitgrenze verwendet wird, beschrieben. 30

#### (A) Verwaltung mit Verarbeitungsdringlichkeit 35

Die Prioritätswert-Bestimmungseinrichtung bestimmt einen Prioritätswert, wie er dazu geeignet ist, ein Programm auszuführen, und zwar unter Berücksichtigung des Funktionsvermögens und der Auslastung jedes Computers und/oder des Dringlichkeitswerts der vom Computer ausgeführten Verarbeitung. Dies ermöglicht es, daß die Verarbeitung im wesentlichen mit derselben Geschwindigkeit ausgeführt wird, unabhängig von Unterschieden des Funktionsvermögens der jeweiligen Computer, und zwar dadurch, daß einem Programm dann ein niedriger Prioritätswert zugeordnet wird, wenn es in einem hochleistungsfähigen Computer abgearbeitet wird, es dagegen einen hohen Prioritätswert erhält, wenn es in einem Computer mit geringer Leistungsfähigkeit abgearbeitet wird. Auf ähnliche Weise kann die Verarbeitung im wesentlichen mit derselben Geschwindigkeit unabhängig von Unterschieden der Auslastung jeweiliger Computer dadurch ausgeführt werden, daß einem stark ausgelasteten Computer hohe Prioritätswerte zugeordnet werden und einem gering ausgelasteten Computer niedrige Prioritätswerte zugeordnet werden. Da der Prioritätswert in jedem Computer auf Grundlage der Verarbeitungsdringlichkeit unter Berücksichtigung des Funktionsvermögens und der Auslastung jedes Computers bestimmt wird, ist es möglich, einen Umkehrungseffekt zu vermeiden, bei dem eine Verarbeitung mit geringerem Prioritätswert früher 40



abgeschlossen wird als eine Verarbeitung mit höherem Prioritätswert. Anders gesagt, können strenger eingehaltene Echtzeiteigenschaften gewährleistet werden als dies bei bekannten Techniken der Fall ist. Die fünf Einrichtungen, die die Prioritätswert-Bestimmungseinrichtung bilden, verfügen jeweils über die folgenden Funktionen.

Die Prioritätswert-Umsetzeinrichtung enthält eine Prioritätswert-Entsprechungstabelle, die für eine Umsetzung zwischen einem Dringlichkeitswert und einem zugehörigen Prioritätswert für die Verarbeitung verwendet wird. Der Inhalt der Prioritätswert-Entsprechungstabelle wird abhängig von Unterschieden des Funktionsvermögens, der Auslastung und des Typs von Computern verändert.

Die Prioritätswert-Veränderungseinrichtung ändert den Prioritätswert eines Programms abhängig vom Prioritätswert, wie er von der Prioritätswert-Umsetzeinrichtung angegeben wird. Diese Prioritätswert-Umsetzeinrichtung gewährleistet in Zusammenarbeit mit der Prioritätswert-Änderungseinrichtung, daß Verarbeitungen mit demselben Dringlichkeitswert mit im wesentlichen derselben Geschwindigkeit von jedem Computer ausgeführt wird, auch wenn diese verschiedene Funktionsvermögen und Auslastungen aufweisen.

Die Computerauswahleinrichtung nimmt auf einen Computer informationsbereich, der aus einer Computerfunktionsvermögen-Informationstabelle und einer Computerauslastung-Informationstabelle besteht, sie wählt einen Computer aus, dem ein Programm mit einem speziellen Dringlichkeitswert zugeordnet wird, um die Auslastungen der Computer im System gleichmäßig auszulasten, und sie sorgt dafür, daß der ausgewählte Computer das Programm ausführt. Die Computerfunktionsvermögen-Informationstabelle gibt die Funktionsvermögen der jeweiligen Computer an, während die Computerauslastung-Informationstabelle die Auslastungen der jeweiligen Computer für jeden Dringlichkeitswert zeigt. Wenn die Auslastung eines Computers schwankt, wird dieses Schwankungsausmaß in der Computerauslastung-Informationstabelle wiedergegeben. Dies verhindert, daß sich eine Anzahl von Verarbeitungen mit hohem Dringlichkeitswert auf einen einzelnen Computer konzentriert, wodurch es möglich ist, in den jeweiligen Computern strenger eingehaltene Echtzeiteigenschaften zu gewährleisten.

Die Programmauswahleinrichtung verwendet eine Programmort-Datenbank, die Information zu Programmen enthält, die denselben Vorgang ausführen, um ein Programm mit einem Dringlichkeitswert, der mit einem vorgegebenen Dringlichkeitswert übereinstimmt, aus einer Gruppe von Programmen auszuwählen, die denselben Vorgang ausführen. Dann wird an das ausgewählte Programm eine Verarbeitungsanforderung ausgegeben. Die Programmort-Datenbank speichert Prioritätswerte oder Dringlichkeitswerte von Programmen, die denselben Vorgang ausführen, und auch die Kennungen der Computer, auf denen diese Programme laufen. So ist es selbst dann möglich, wenn die oben genannte Prioritätswert-Änderungseinrichtung nicht verwendet wird, mehrere Programme mit verschiedenen Dringlichkeitswerten zu erstellen, das Programm mit dem geeignetsten Dringlichkeitswert aus diesen Programmen zur Ausführung der Verarbeitung aufzufordern, und dafür zu sorgen, daß das aufgeforderte Programm die Verarbeitung mit gewünschter Geschwindigkeit ausführt.

Wenn ein gerade eine Verarbeitung ausführendes

Programm dazu aufgefordert wird, eine andere Verarbeitung auszuführen, fragt die Verarbeitungsverlegeteinrichtung aus der Programmauswahleinrichtung die Orte der anderen Programme ab, die denselben Vorgang wie das angeforderte Programm ausführen, und sie verlegt die Verarbeitungsanforderung auf ein Programm, das im Ergebnis durch die Programmauswahleinrichtung spezifiziert wurde. So kann die neu angeforderte Verarbeitung für das aktuell laufende Programm durch ein anderes Programm ausgeführt werden, das die Verarbeitung mit im wesentlichen derselben Geschwindigkeit (mit demselben Dringlichkeitswert) ausführen kann, ohne daß auf den Abschluß der aktuell ausgeführten Verarbeitung durch das angeforderte Programm gewartet werden muß, wodurch strenger eingehaltene Echtzeiteigenschaften gewährleistet werden.

#### (B) Verwaltung mit Verarbeitungszeitgrenze

Die Prioritätswert-Bestimmungseinrichtung bestimmt einen Prioritätswert, wie er für die Ausführung eines Programms geeignet ist, und zwar unter Berücksichtigung des Funktionsvermögens und der Auslastung jedes Computers und/oder eines Wertesatzes aus einer Zeitgrenze und einer Verarbeitungsmenge für die vom Programm ausgeführte Verarbeitung. Auf diese Weise wird der Satz aus der Zeitgrenze und der Verarbeitungsmenge für die Verarbeitung in einen örtlichen Prioritätswert umgesetzt, wie er für jeden Computer definiert ist, und zwar abhängig vom Funktionsvermögen und der Auslastung des Computers, und die Verarbeitung wird mit dem umgesetzten Prioritätswert ausgeführt. Dadurch ist es möglich, eine neu angeforderte Verarbeitung mit kürzerer Zeitgrenze auszuführen, bevor eine zuvor zugeordnete Verarbeitung mit längerer Zeitgrenze ausgeführt wird. Auch wird, da die Zeitgrenze für die Verarbeitung in einen örtlichen Prioritätswert für einen Computer unter Berücksichtigung des Funktionsvermögens und der Auslastung des Computers umgesetzt werden kann, dann, wenn sich die verstrichene Zeit der Zeitgrenze für die Verarbeitung nähert, wie sie bereits dem Computer zugeordnet ist, die Zeitgrenze für die Verarbeitung verkürzt werden, und es wird erneut ein zugehöriger Prioritätswert abhängig von der verkürzten Zeitgrenze für die Verarbeitung bestimmt, wodurch eine neu angeforderte Verarbeitung die Ausführung der zuvor zugeordneten Verarbeitung nicht behindert. Anders gesagt, können strenger eingehaltene Echtzeiteigenschaften gewährleistet werden als beim Stand der Technik. Die acht Einrichtungen, die die Prioritätswert-Bestimmungseinrichtung bilden, haben jeweils die folgenden Funktionen.

Die Prioritätswert-Umsetzeinrichtung enthält eine Prioritätswert-Entsprechungstabelle, die die Entsprechung zwischen der Zeitgrenze pro Einheitsverarbeitungs menge und einem Prioritätswert angibt, wie dazu verwendet, eine Umsetzung zwischen einem Satz aus einer Zeitgrenze und einer Verarbeitungsmenge für die Verarbeitung und einem zugehörigen Prioritätswert vorzunehmen. Der Inhalt der Prioritätswert-Entsprechungstabelle wird abhängig von Änderungen des Funktionsvermögens, der Auslastung und des Typs von Computern verändert.

Die Prioritätswert-Änderungseinrichtung ändert den Prioritätswert eines Programms abhängig von einem Prioritätswert, wie er von der Prioritätswert-Umsetzeinrichtung angegeben wird. Diese Prioritätswert-Umsetzeinrichtung ermöglicht in Zusammenarbeit mit



der Prioritätswert-Änderungseinrichtung, daß eine Verarbeitung mit einem örtlichen Prioritätswert für jeden Computer entsprechend der der Verarbeitung auferlegten Zeitgrenze ausgeführt wird. Daher ist es möglich, eine Verarbeitung mit kürzerer Zeitgrenze vor einer anderen Verarbeitung mit längerer Zeitgrenze auszuführen, wodurch strengeres Einhalten von Echtzeiteigenschaften für jede Verarbeitung im verteilten Computersystem gewährleistet ist. Wenn eine Verarbeitung mit langer Zeitgrenze nicht abgearbeitet wird, um bevorzugt andere Verarbeitungen mit kurzer Zeitgrenze auszuführen, nähert sich die abgelaufene Zeit der Zeitgrenze der Verarbeitung, die noch nicht abgearbeitet ist. In diesem Fall wird der Prioritätswert der Verarbeitung abhängig von der verkürzten zur Verfügung stehenden Zeit für die Verarbeitung auf einen größeren Wert korrigiert, so daß die Verarbeitung mit dem korrigierten Prioritätswert ausgeführt werden kann.

Die Verbindungszeit-Vorhersageeinrichtung verfügt über eine Vorhersagezeittabelle zum Abspeichern einer Verbindungszeit für jeden Computer, und sie gibt Information über die Zeit, wie sie für eine Verbindung mit einem bestimmten Computer erforderlich ist, wenn eine entsprechende Anfrage erfolgt. Auf diese Weise kann die Zeitgrenze für die Verarbeitung genauer eingestellt werden, wenn die Verbindungszeit zwischen Computern nicht vernachlässigbar ist.

Die Computerauswahleinrichtung nimmt auf einen Computer informationsbereich Bezug, der aus einer Computerfunktionsvermögen-Informationstabelle und einer Computerauslastung-Informationstabelle besteht, sie wählt einen Computer so aus, daß die Auslastung oder Verarbeitung mit vorgegebener Zeitgrenze pro Einheitsverarbeitungs menge gleich auf die Computer verteilt ist, und sie sorgt dafür, daß der ausgewählte Computer ein Programm ausführt. Die Computerfunktionsvermögen-Informationstabelle zeigt die Funktionsvermögen der jeweiligen Computer, während die Computerauslastung-Informationstabelle die Auslastungen der jeweiligen Computer für jeden Dringlichkeitswert zeigt. Wenn sich die Auslastung eines Computers ändert, wird das Änderungsausmaß in der Computerauslastung-Informationstabelle wiedergegeben. Dies verhindert, daß eine Anzahl von Verarbeitungen mit kurzer Zeitgrenze auf einen einzelnen Computer konzentriert wird, wodurch es möglich ist, Echtzeiteigenschaften in den jeweiligen Computern strenger einzuhalten.

Die Programmauswahleinrichtung verwendet eine Programmort-Datenbank, die Information zu Programmen enthält, die denselben Vorgang ausführen, um ein Programm mit einer Zeitgrenze pro Verarbeitungsmengeeinheit, die mit der spezifizierten übereinstimmt, aus einer Gruppe von Programmen auszuwählen, die denselben Vorgang ausführen. Eine Verarbeitungsanforderung wird dann an das ausgewählte Programm ausgegeben. Die Programmort-Datenbank speichert entweder Prioritätswerte oder Dringlichkeitswerte für Programme ein, die denselben Vorgang ausführen, und sie speichert die Kennungen von Computern ab, auf denen diese Programme laufen. Demgemäß ist es selbst dann, wenn die oben genannte Prioritätswert-Änderungseinrichtung nicht verwendet wird, möglich, mehrere Programme mit verschiedenen Zeitgrenzen bereitzustellen, das Programm mit der am besten geeigneten Zeitgrenze, wie aus diesen Programmen ausgewählt, zum Ausführen der Verarbeitung aufzufordern, und dafür zu sorgen, daß das angeforderte Programm die Verarbeitung

mit gewünschter Geschwindigkeit ausführt.

Wenn ein gerade eine Verarbeitung ausführendes Programm dazu aufgefordert wird, eine andere Verarbeitung auszuführen, fragt die Verarbeitungsverlegungseinrichtung bei der Programmauswahleinrichtung die Orte anderer Programme ab, die denselben Vorgang wie das angeforderte Programm ausführen, und sie verlegt die Verarbeitungsanforderung auf ein Programm, das im Ergebnis durch die Programmauswahleinrichtung spezifiziert wurde. So kann eine neue Verarbeitung, zu der das aktuell arbeitende Programm aufgefordert wird, durch ein anderes Programm ausgeführt werden, das die Verarbeitung mit im wesentlichen derselben Geschwindigkeit ausführen kann, ohne daß auf den Abschluß der aktuell ausgeführten Verarbeitung durch das angeforderte Programm gewartet werden muß, wodurch strengeres Einhalten von Echtzeiteigenschaften gewährleistet ist.

Die Verarbeitungszeit-Voraussageeinrichtung berechnet die Zeit, wie sie dazu erforderlich ist, ein Programm abzuschließen, und zwar gestützt auf die restliche Verarbeitungs menge, wie vom Programm auszuführen, und gestützt auf eine Zeitgrenze pro Einheitsverarbeitungs menge des Programms.

Die Zeitgrenze-Einstelleinrichtung bestimmt eine neue Zeitgrenze für die Verarbeitung gestützt auf eine Zeitgrenze für die Verarbeitung, wie beim Anfordern der Verarbeitung spezifiziert, und gestützt auf die Zeit, wie sie ab dem Start der Ausführung der Verarbeitung verstrichen ist. Die Zeitgrenze-Einstelleinrichtung bestimmt in Zusammenwirkung mit der Verarbeitungszeit-Voraussageeinrichtung, ob die Zeitgrenze für die Verarbeitung erfüllt wird, wenn ein Programm mit dem aktuellen Prioritätswert abgearbeitet wird. Wenn erkannt wird, daß die Zeitgrenze nicht erfüllt wird, kann eine kürzere Zeitgrenze eingestellt werden. Unter Verwendung der neu eingestellten Zeitgrenze können die Prioritätswert-Umsetzeinrichtung und die Prioritätswert-Änderungseinrichtung dem Programm einen höheren Prioritätswert verleihen und die Verarbeitung an ein anderes Programm verlegen, das denselben Vorgang ausführt und der neuen Zeitgrenze genügen kann, damit dieses Programm die Verarbeitung vornimmt. Auf diese Weise kann die Verarbeitung so verwaltet werden, daß der ursprünglich vergebenen Zeitgrenze genügt wird.

Die Erfindung wird im folgenden anhand von durch Figuren veranschaulichten Ausführungsbeispielen näher beschrieben.

Fig. 1 ist ein Diagramm, das die Konfiguration eines verteilten Computersystems gemäß einem ersten Ausführungsbeispiel der Erfindung zeigt;

Fig. 2 zeigt eine Datenstruktur einer Prioritätswert-Entsprechungstabelle, die die Entsprechung zwischen Dringlichkeitswerten und Prioritätswerten angibt;

Fig. 3 zeigt ein anderes Beispiel zum Realisieren einer Prioritätswert-Umsetzeinrichtung;

Fig. 4 zeigt eine Form eines Programms zum Ausgeben einer Verarbeitungsanforderung mit einem spezifizierten Dringlichkeitswert;

Fig. 5 ist ein Diagramm, das einen Verarbeitungsablauf zum Ausführen von Verbindungen zum Spezifizieren eines Dringlichkeitswerts zeigt;

Fig. 6 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Ändern des Inhalts einer Prioritätswert-Entsprechungstabelle repräsentiert, die die Entsprechung zwischen Dringlichkeitswerten und Prioritätswerten abhängig von sich ändernder Auslastung angibt;



Fig. 7 zeigt, wie die Prioritätswert-Entsprechungstabelle abhängig von einer Zunahme oder Abnahme der Auslastung geändert wird;

Fig. 8 zeigt ein Programm zum Modifizieren der Prioritätswert-Entsprechungstabelle abhängig von einer Zunahme oder Abnahme der Auslastung;

Fig. 9 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Ändern des Prioritätswerts eines Programms in Übereinstimmung mit einer Modifizierung der Prioritätswert-Entsprechungstabelle zeigt;

Fig. 10 ist ein Diagramm, das ein zweites Ausführungsbeispiel der Erfindung und einen Verarbeitungsablauf zum Ausführen von Verbindungen zum Spezifizieren eines Dringlichkeitswerts bei diesem Ausführungsbeispiel zeigt;

Fig. 11 zeigt die Konfiguration eines dritten Ausführungsbeispiels der Erfindung;

Fig. 12 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Starten eines Programms beim dritten Ausführungsbeispiel zeigt;

Fig. 13 zeigt Beispiele für eine Computerauslastungs-Informationstabelle, eine Auslastungswerttabelle und eine Auslastung/Funktionsvermögen-Tabelle beim dritten Ausführungsbeispiel der Erfindung;

Fig. 14 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Verschieben eines Programms repräsentiert;

Fig. 15 zeigt andere Beispiele für die Computerauslastungs-Informationstabelle, die Auslastungswerttabelle und die Auslastung/Funktionsvermögen-Tabelle beim dritten Ausführungsbeispiel der Erfindung;

Fig. 16 und 17 sind Diagramme, die die Konfiguration eines vierten bzw. fünften Ausführungsbeispiels der Erfindung zeigen;

Fig. 18 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Auswählen eines Programms beim fünften Ausführungsbeispiel der Erfindung veranschaulicht;

Fig. 19 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Herausfinden eines Programms mit speziellem Prioritätswert veranschaulicht;

Fig. 20 ist ein Diagramm, das die Konfiguration eines sechsten Ausführungsbeispiels der Erfindung und einen Verarbeitungsablauf zum Auswählen eines Programms beim sechsten Ausführungsbeispiel zeigt;

Fig. 21 ist ein Diagramm, das die Konfiguration eines siebten Ausführungsbeispiels der Erfindung und einen Verarbeitungsablauf zum Verlegen einer Verarbeitungsanforderung beim siebten Ausführungsbeispiel zeigt;

Fig. 22 ist ein Diagramm, das die Konfiguration eines achten Ausführungsbeispiels der Erfindung zeigt;

Fig. 23 zeigt eine Datenstruktur einer Prioritätswert-Entsprechungstabelle, die die Entsprechung zwischen einer Zeitgrenze und einem Prioritätswert angibt;

Fig. 24 zeigt ein Programm zum Herleiten eines Prioritätswerts aus einer vorgegebenen Zeitgrenze;

Fig. 25 zeigt eine Form eines Programms zum Ausgeben einer Verarbeitungsanforderung mit spezifizierter Zeitgrenze;

Fig. 26 ist ein Diagramm, das einen Verarbeitungsablauf zum Ausführen von Verbindungen zum Spezifizieren einer Zeitgrenze beim achten Ausführungsbeispiel zeigt;

Fig. 27 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Modifizieren des Inhalts der Prioritätswert-Entsprechungstabelle abhängig von sich ändernder Auslastung veranschaulicht;

Fig. 28 zeigt, wie die Prioritätswert-Entsprechungstabelle abhängig von einer Zunahme oder Abnahme der Auslastung modifiziert wird;

Fig. 29 ist ein Diagramm, das ein neuntes Ausführungsbeispiel der Erfindung und einen Verarbeitungsablauf zum Ausführen von Verbindungen zum Spezifizieren einer Zeitgrenze unter Berücksichtigung der Verbindungszeit beim neunten Ausführungsbeispiel zeigt;

Fig. 30 ist ein Diagramm, das die Konfiguration eines zehnten Ausführungsbeispiels zeigt;

Fig. 31 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Starten eines Programms beim zehnten Ausführungsbeispiel veranschaulicht;

Fig. 32 zeigt Beispiele einer Computerauslastungs-Informationstabelle, einer Auslastungswerttabelle und einer Auslastung/Funktionsvermögen-Tabelle beim zehnten Ausführungsbeispiel der Erfindung;

Fig. 33 ist ein Flußdiagramm, das einen Verarbeitungsablauf zum Verschieben eines Programms beim zehnten Ausführungsbeispiel der Erfindung veranschaulicht;

Fig. 34 zeigt andere Beispiele der Computerauslastungs-Informationstabelle, der Auslastungswerttabelle und der Auslastung/Funktionsvermögen-Tabelle beim zehnten Ausführungsbeispiel der Erfindung;

Fig. 35 ist ein Diagramm, das die Konfiguration eines elften Ausführungsbeispiels der Erfindung zeigt;

Fig. 36 zeigt einen Verarbeitungsablauf zum Auswählen eines Programms beim elften Ausführungsbeispiel der Erfindung;

Fig. 37 ist ein Diagramm, das die Konfiguration eines zwölften Ausführungsbeispiels der Erfindung und einen Verarbeitungsablauf zum Auswählen eines Programms beim zwölften Ausführungsbeispiel zeigt; und

Fig. 38 ist ein Diagramm, das die Konfiguration eines dreizehnten Ausführungsbeispiels der Erfindung und einen Verarbeitungsablauf zum Verlegen einer Verarbeitungsanforderung beim dreizehnten Ausführungsbeispiel zeigt.

Zunächst wird eine Reihe von Ausführungsbeispielen unter Verwendung des Dringlichkeitskonzepts dargestellt, gefolgt von Ausführungsbeispielen, die das Zeitgrenzenkonzept verwenden.

#### (A) System unter Verwendung der Verarbeitungsdringlichkeit

Fig. 1 zeigt die Konfiguration eines verteilten Computersystems gemäß einem ersten Ausführungsbeispiel der Erfindung. In diesem System sind mehrere Computer 100–103 mit einem Netzwerk 110 in solcher Weise verbunden, daß Programme auf den jeweiligen Computern in Zusammenarbeit miteinander ablaufen. Während im System von Fig. 1 nur vier miteinander verbundene Computer dargestellt sind, kann zum Aufbauen eines Systems eine Anzahl von Computern verwendet werden, die größer oder kleiner ist als diese Zahl. Das Netzwerk 110 kann einen Übertragungsbus, optische Fasern, ein Kreuzschienenschalter-Vermittlungssystem usw. enthalten. Programme auf den jeweiligen Computern stehen über das Netzwerk 110 miteinander in Verbindung. Einzelheiten zur internen Konfiguration sind nur für die Computer 100, 101 dargestellt. Jeder der Computer umfaßt eine zentrale Verarbeitungseinheit 120, eine Hauptspeichereinheit 121, eine Eingabe/Ausgabe-Steuereinheit 122 usw., und empfängt und sendet Daten über einen internen Bus 125. Diese Computer





können sich hinsichtlich des Funktionsvermögens der zentralen Verarbeitungseinheit 120 oder dergleichen voneinander unterscheiden. Die Eingabe/Ausgabe-Steuereinheit 122 in jedem Computer ist im allgemeinen mit einer externen Speichereinheit 123 und einem Terminal 124 verbunden, um Daten von außen zu empfangen und nach außen zu senden. Die Hauptspeichereinheiten 121 enthalten Benutzerprogramme 130—136, Prioritätsschlangen 140, 141, Scheduler 142, 143, Prioritätswert-Umsetzeinrichtungen 144, 145 und Prioritätswert-Änderungseinrichtungen 146, 147, die alle durch die zentralen Verarbeitungseinheiten 120 der jeweiligen Computer betrieben werden. Im allgemeinen ist jeder Computer mit seinem eigenen Betriebssystem versehen, um die Ausführungen von Programmen und andere Bedingungen im zugehörigen Computer zu verwalten. Die Prioritätsschlange und der Scheduler sind häufig in einem Teil des Betriebssystems enthalten. In Fig. 1 ist die Beschreibung des Betriebssystems weggelassen. Die Anzahl von Benutzerprogrammen und die Art ihrer Anordnung in den jeweiligen Computern können vom Benutzer beliebig festgelegt werden.

Wenn mehrere Programme miteinander zusammenwirkend in einem Mehrprozessor-Computersystem oder einem verteilten Computersystem, wie in Fig. 1 dargestellt, laufen, werden die Benutzerprogramme in den jeweiligen Computern normalerweise in ihrer Bearbeitungsreihenfolge verwaltet, wie sie durch Reihenfolgeschlangen im Betriebssystem oder dergleichen angegeben ist. Jedoch kann sich die Bearbeitungsdringlichkeit von einem Benutzerprogramm zum anderen unterscheiden. Aus diesem Grund ist jedes Benutzerprogramm mit einem örtlichen Prioritätswert in jedem Computer in solcher Weise versehen, daß die Benutzerprogramme abhängig vom vorgegebenen Prioritätswert in eine Schlange eingegeben werden, um die Reihenfolge der Ausführung dieser Benutzerprogramme zu verwalten. Die Prioritätsschlangen 140, 141 bilden die Schlangen für die jeweiligen Prioritätswerte in den Computern 100, 101. Bei dem in Fig. 1 dargestellten Ausführungsbeispiel ist angenommen, daß der Computer 100 Prioritätswerte im Bereich von 0—255 hat, während der Computer 101 Prioritätswerte im Bereich von 0—127 hat. Für jeden der Prioritätswerte ist eine Schlange gebildet, mittels der die Scheduler 142, 143 das Programm bestimmen, das als nächstes auszuführen ist. Die Prioritätsschlangen bestehen aus Programmstrukturen (Beispiele für eine Programmstruktur sind in Fig. 1 als Blöcke 160—166 dargestellt), von denen jede Information zu einem zugehörigen Benutzerprogramm enthält. Während Programmstrukturen für Programme D, E und F usw. in den Prioritätsschlangen existieren, die von den Benutzerprogrammen 130—136 abweichen, die in der Hauptspeichereinheit 131 liegen, liegen diese Programme nicht in der Hauptspeichereinheit 121 sondern in der externen Speichereinheit 123. Diese Programme D, E, F usw. können ausgeführt werden, nachdem ihnen zugeordnete, erforderliche Programmdateien von der externen Speichereinheit 123 in die Hauptspeichereinheit 121 überschrieben sind.

Um das verteilte Computersystem dazu aufzufordern, eine Verarbeitung auszuführen, gibt jeder Benutzer dieses verteilten Computersystems an einen der Computer in diesem System die Anweisung, sein Benutzerprogramm neu abzuarbeiten, oder ergibt dem System die Anweisung, ein bereits vorhandenes Programm abzuarbeiten, wozu er Daten über eine Eingabe-einrichtung eingibt. Jedes der Programme im verteilten

Computersystem führt seine Verarbeitung aus, während es nach Bedarf mit anderen Programmen in Verbindung steht, und es liefert die Ergebnisse an den Benutzer zurück, oder es zeigt sie auf dem Terminal eines der Computer an. Anders gesagt, entspricht das Eingeben von Information in ein verteiltes Computersystem dem Eingeben eines Programms in ein verteiltes Computersystem oder dem Eingeben von Daten für ein Programm über eine Eingabeeinrichtung wie ein Terminal oder dergleichen. Auch entspricht die Ausgabe aus dem verteilten Computersystem der Ausgabe von Ergebnissen eines Programms, wie vom verteilten Computersystem ausgeführt, als Anzeige auf einem Schirm eines Terminals oder dergleichen.

Die Scheduler 142, 143 bestimmen jeweils das als nächstes auszuführende Programm abhängig vom Inhalt der jeweiligen Prioritätsschlangen 140 bzw. 141. Bei diesem Ausführungsbeispiel beinhaltet die Prioritätsschlange 140 des Computers 100 ein Programm A (Benutzerprogramm 130) in einer Schlange für den Wert 0, und Programme B (Benutzerprogramm 131) und C (Benutzerprogramm 132) in einer Schlange für den Wert 1. In dieser Situation werden dann, wenn der Scheduler 142 so ausgebildet ist, daß er Programme ausgehend von einem solchen mit kleinstem Prioritätswert ausführt, die Programme in der Reihenfolge A, B, C, D, E, F, ... ausgeführt, abhängig von der so ausgebildeten Prioritätsschlange. Die Prioritätsschlangen 140, 141 und die Scheduler 142, 143 bilden normalerweise einen Teil des Betriebssystems eines jeweiligen Computers. Die Vorgehensweise, wie die Reihenfolge zur Ausführung von Programmen bestimmt wird, kann sich von der oben angegebenen unterscheiden, und zwar abhängig vom Typ des verwendeten Betriebssystems. Das Ausführungsbeispiel wird nachfolgend unter der Annahme beschrieben, daß ein kleinerer Prioritätswert höhere Priorität anzeigt (d. h., daß ein Programm mit kleinerem Prioritätswert früher ausgeführt wird).

Wie oben angegeben, können Programme mit verschiedenen Dringlichkeiten auf gleichmäßige Weise in ein und demselben Computer verwaltet werden, wenn jedem Programm ein Prioritätswert angesichts der Ausführungsreihenfolge für Programme zugeordnet wird. Jedoch sind die örtlichen Prioritätsschemata nicht miteinander verträglich, und zwar wenn es um Computer mit verschiedenen Funktionsvermögen, Auslastungen und Typen geht. Um mit dieser Schwierigkeit fertig zu werden, führt dieses Ausführungsbeispiel die "Dringlichkeit" ein, bei der es sich um ein gemeinsames Konzept im gesamten verteilten Computersystem handelt, um ein globales Prioritätsschema für das gesamte verteilte Computersystem einschließlich mehrerer Computer mit voneinander verschiedenen Funktionsvermögen, Auslastungen und Typen gleichmäßig zu verwalten. Die Dringlichkeit kann als globales Prioritätsschema angesehen werden, das auf das gesamte verteilte Computersystem anwendbar ist. Bei diesem Ausführungsbeispiel ist angenommen, daß die Dringlichkeit in einem Bereich 0—31 spezifiziert werden kann. Einrichtungen bei diesem Ausführungsbeispiel, die zum Einführen der Dringlichkeit dienen, sind die Prioritätswert-Umsetzeinrichtungen 144, 145 und der Prioritätswert-Änderungseinrichtungen 146, 147. Die Beschreibung des Ausführungsbeispiels geht nachfolgend unter der Annahme weiter, daß ein kleinerer Dringlichkeitswert einen höheren Dringlichkeitswert repräsentiert, ähnlich wie dies für den Prioritätswert angenommen ist.

Die Prioritätswert-Umsetzeinrichtungen 144, 145 sind



Einrichtungen zum Umsetzen des Dringlichkeitswerts, bei dem es sich um einen gemeinsamen Datenwert im verteilten Computersystem handelt, in einen örtlichen Prioritätswert, bei dem es sich um einen Datenwert handelt, wie er jedem Computer zu eigen ist und umgekehrt. Bei diesem Ausführungsbeispiel sind die Prioritätswert-Umsetzungseinrichtungen 144, 145 mit Prioritätswert-Entsprechungstabellen 150, 151 versehen, die die Entsprechung zwischen dem Dringlichkeitswert (UR) und dem örtlichen Prioritätswert (PRI) in jedem Computer angeben. Aus Fig. 1 ist erkennbar, daß der Dringlichkeitswert im Computer 100 dem Prioritätswert 0 oder 1 entspricht, und daß der Dringlichkeitswert 1 den Prioritätswerten 2—4 entspricht, wohingegen im Computer 101, der schlechteres Funktionsvermögen als der Computer 100 hat, der Dringlichkeitswert dem Prioritätswert 0 entspricht, während der Dringlichkeitswert 1 dem Prioritätswert 1 entspricht. Auf diese Weise ist im Computer 101 demselben Dringlichkeitswert ein höherer Prioritätswert zugeordnet als im Computer 100. Dies stellt sicher, daß ein Programm mit derselben Dringlichkeit in den beiden Computern 100, 101 mit verschiedenen Funktionsvermögen im wesentlichen mit derselben Geschwindigkeit ausgeführt werden kann. Die Inhalte der Prioritätswert-Entsprechungstabellen 150, 151 können abhängig von den Auslastungswerten der jeweiligen Computer so modifiziert werden, daß derselbe Dringlichkeitswert in einem stark ausgelasteten Computer einem höheren Prioritätswert entspricht als in einem schwach ausgelasteten Computer. Die Prioritätswert-Umsetzungseinrichtungen 144, 145 benutzen die jeweiligen Prioritätswert-Entsprechungstabellen 150, 151, um einen vorgegebenen Dringlichkeitswert in einen entsprechenden Prioritätswert für die zugeordneten Computer 100, 101 umzusetzen, und um einen Prioritätswert in einen entsprechenden Dringlichkeitswert umzusetzen.

Während die Prioritätswert-Entsprechungstabellen 150, 151 auf Grundlage der in Fig. 1 dargestellten Feldstrukturen erzeugt werden können, kann ein Dringlichkeitswert mehreren Prioritätswerten entsprechen, so daß die Umsetzung von einem Dringlichkeitswert in einen Prioritätswert oder mehrere nicht auf wirkungsvolle Weise ausgeführt werden kann. Fig. 2 zeigt ein Beispiel für eine in der Sprache C geschriebene Datenstruktur für die Prioritätswert-Entsprechungstabelle 150, die dazu dient, die Umsetzung zwischen Dringlichkeitswerten und Prioritätswerten wirkungsvoll auszuführen. Um die Umsetzung zwischen Dringlichkeitswerten und Prioritätswerten zu erleichtern, besteht die Prioritätswert-Entsprechungstabelle aus einer Kombination zweier Felder. In dieser Prioritätswert-Entsprechungstabelle 150 kann, da ein Prioritätswert in eindeutiger Weise in einen entsprechenden Dringlichkeitswert umgesetzt werden kann, ein Feld 170 verwendet werden. Da jedoch ein Dringlichkeitswert in mehrere entsprechende Prioritätswerte umgesetzt werden kann, ist zum Abspeichern des kleinsten Prioritätswerts und des größten Prioritätswerts für jeden Dringlichkeitswert eine andere Datenstruktur verwendet (Feld 171). Ein Wert, wie er von jedem Feldelement angegeben wird, ist in einer Datenstruktur 172 dargestellt. Wenn diese Datenstruktur verwendet wird, kann die Umsetzung zwischen einem Dringlichkeitswert u und einem Prioritätswert p auf die folgende Weise erfolgen:

(1) Umsetzung von einem Prioritätswert in einen Dringlichkeitswert:

u-to\_urgency[p].

(2) Umsetzung von einem Dringlichkeitswert in einem Prioritätswert:

p-to\_ any integer value between priority[u]  
.start and topriority[u].end

Bei diesem Ausführungsbeispiel ist die Beziehung zwischen einem Dringlichkeitswert und einem Prioritätswert auf eine Einfach-Mehrfach-Beziehung fixiert. Daher kann, während die Umsetzung von einem Prioritätswert in einen Dringlichkeitswert durch ein einfaches Feld realisiert ist, die Entsprechung zwischen einem Dringlichkeitswert und einem Prioritätswert in einer Einfach-Mehrfach-Beziehung oder einer Mehrfach-Mehrfach-Beziehung wiedergegeben sein, wenn der Bereich der Dringlichkeitswerte größer ist als der bei diesem Ausführungsbeispiel dargestellte Bereich. Wenn eine Mehrfach-Mehrfach-Beziehung verwendet wird, muß das Feld 170 in eine ähnliche Struktur umgewandelt werden, wie sie das Feld 171 aufweist. Bei einer Einfach-Mehrfach-Beziehung kann eine Prioritätswert-Entsprechungstabelle für diesen Fall dadurch realisiert werden, daß das Feld 170 durch das Feld 171 ersetzt wird. Es ist zu beachten, daß zum Ändern des Bereichs der Prioritätswerte oder des Bereichs der Dringlichkeitswerte die Anzahlen der Elemente in den Feldern 170, 171 verändert werden müssen.

Während beim in den Fig. 1 und 2 dargestellten Beispiel die Prioritätswert-Entsprechungstabellen 150, 151 dazu verwendet werden, einen Dringlichkeitswert in einen Prioritätswert umzusetzen, und umgekehrt, kann die Umsetzung auch ohne Verwendung derartiger Prioritätswert-Entsprechungstabellen ausgeführt werden. Fig. 3 zeigt den Inhalt einer Prioritätswert-Umsetzungseinrichtung 144, die keine Prioritätswert-Entsprechungstabelle verwendet. Der Ablauf ist in der Sprache C geschrieben. Ein Block 180 entspricht einem Umsetzablauf von einem Dringlichkeitswert in einen Prioritätswert, und ein Block 181 entspricht einem Umsetzablauf von einem Prioritätswert in einen Dringlichkeitswert. Während dieses Ausführungsbeispiel so ausgewählt ist, daß optimale Gleichung eine solche verwendet wird, die ausdrückt, daß ein Prioritätswert proportional zum Quadrat eines Dringlichkeitswerts ist, ist es auch möglich, empirisch eine optimale Beziehung für jeden Computer im verteilten Computersystem herzuleiten und die hergeleitete Beziehung in Form einer Gleichung erster Ordnung oder einer Gleichung höherer Ordnung wiederzugeben. Dieses Ausführungsbeispiel verwendet das Computerfunktionsvermögen pro Einheitsauslastung als Proportionalitätskoeffizient. Auch ist es bei diesem Ausführungsbeispiel geschickt, daß eine Auslastung durch eine Anzahl von Programmen in der Prioritätsschlange (die Anzahl ausführbarer Programme) und das Computerfunktionsvermögen, d. h. das CPU(zentrale Verarbeitungseinheit)-Funktionsvermögen in der Einheit von 10 MIPS (1 MIPS repräsentiert das Funktionsvermögen einer CPU, die dazu in der Lage ist, eine Million Anweisungen pro Sekunde auszuführen) repräsentiert ist. Unter Verwendung dieses Umsetzablaufs kann ein niedrigerer Prioritätsrang (ein höherer Prioritätswert) für denselben Dringlichkeitswert eingestellt werden, wenn das CPU-Funktionsvermögen pro Programm höher ist, während ein höherer Prioritätsrang (ein kleinerer Prioritätswert) für denselben Dringlichkeitswert eingestellt werden kann, wenn das CPU-



Funktionsvermögen pro Programm kleiner ist.

Die Prioritätsrang-Änderungseinrichtungen 146, 147 in Fig. 1 ändern jeweils den Prioritätsrang jedes Programms abhängig von einem örtlichen Prioritätsschema im zugehörigen Computer 100, 101, wie durch die Prioritätsrang-Umsetzungseinrichtungen 144 bzw. 145 berechnet. Z.B. verfügt in Fig. 1 ein Programm J (Benutzerprogramm 133) über einen Dringlichkeitswert 1 und einen Prioritätswert 4. Wenn angenommen wird, daß sich der Dringlichkeitswert dieses Programms J von 1 auf 0 ändert, setzt die Prioritätswert-Umsetzungseinrichtung 144 den geänderten Dringlichkeitswert in einen entsprechenden Prioritätswert um, um den neuen Prioritätswert 0 (oder 1) zu erhalten. Die Prioritätswert-Änderungseinrichtung 146 verwendet diesen Wert, um den Prioritätswert des Benutzerprogramms auf 0 zu ändern. Gleichzeitig wird eine Programmstruktur 163 für das Programm J in die Schlange für den Wert 0 verlegt.

Die Prioritätswert-Umsetzungseinrichtungen 144, 145 sowie die Prioritätswert-Änderungseinrichtungen 146, 147 sind beide Dienstprogramme, die in Zusammenwirkung mit einem Teil der jeweiligen Betriebssysteme oder allen arbeiten. Um diese Abläufe, wie solche Dienstprogramme, aufzubauen, ist es erforderlich, eine solche Form zu verwenden, daß die Prioritätsrang-Änderungseinrichtungen 146, 147 die jeweiligen Betriebssysteme dazu auffordern, den Prioritätswert eines Programms zu ändern. Jedes der aufgeforderten Betriebssysteme ändert dann den Prioritätswert eines Programms und verlegt die Programmstruktur auf eine Schlange für den geänderten Prioritätswert.

Nachfolgend wird ein Verfahren zum Anfordern von Verarbeitung für Computer gestützt auf die Dringlichkeit erläutert, wobei der Prioritätswert-Umsetzablauf und der Prioritätswert-Änderungsablauf verwendet werden, wie sie bei diesem Ausführungsbeispiel beschrieben sind. Um eine Verarbeitung zwischen Computern in einem verteilten Computer-Echtzeitsystem anzufordern, kann die Systemarchitektur dadurch vereinfacht werden, daß ein Dringlichkeitswert angegeben wird, wie er für eine spezielle Verarbeitung gewünscht ist. Um diese Vorgehensweise zu verwenden, ermöglicht es die Erfindung, einem anfordernden Programm, einen Dringlichkeitswert zu spezifizieren, wenn es eine Verarbeitung anfordert. Fig. 4 zeigt ein Beispiel für ein Verfahren zum Spezifizieren eines Dringlichkeitswerts, geschrieben in der Sprache C, wobei ein Block 190 ein Verfahren zum Spezifizieren eines Dringlichkeitswerts im ersten Argument einer Verarbeitungsanforderung repräsentiert, während ein Block 191 ein Verfahren zum Spezifizieren eines Dringlichkeitswerts im letzten Argument einer Verarbeitungsanforderung repräsentiert. Übrigens muß das anfordernde Programm nicht direkt angeben, wie ein Dringlichkeitswert zu spezifizieren ist, sondern statt dessen können ein Compiler oder Übersetzer dazu verwendet werden, eine Übersetzung in die Formen vorzunehmen, wie sie durch die Blöcke 190, 191 repräsentiert sind.

Fig. 5 zeigt schematisch einen Verarbeitungsablauf zum Ausgeben einer Verarbeitungsanforderung unter Verwendung des oben angegebenen Spezifizierungsverfahrens für den Dringlichkeitswert. Elemente 100, 101, 110, 130—136, 140—147, 150—151 sowie 160—161 in Fig. 5 stimmen mit Elementen überein, wie sie in Fig. 1 mit denselben Bezugszahlen gekennzeichnet sind. Eine Veranschaulichung der Computer 102, 103 und Einzelheiten innerhalb der Computer sind in Fig. 5 weggelassen. Dieses Ausführungsbeispiel zeigt einen Ablauf, wie er aus-

geführt wird, wenn ein Benutzerprogramm 133 eine Verarbeitungsanforderung an ein anderes Benutzerprogramm 136 ausgibt. Verbindungen, wie sie durch Pfeile 200—206 repräsentiert sind, beschreiben den Ablauf der Verarbeitungsschritte. In Fig. 5 ist angenommen, daß Daten, die jeweils an einen eine Verbindung kennzeichnenden Pfeil geschrieben sind, wichtige Daten angeben, wie sie über diese Verbindung übertragen werden. Das Benutzerprogramm 133 hat den Prioritätswert 4 (Programmstruktur 163) in der Prioritätsschlange 140. Zunächst fordert das Benutzerprogramm 133 eine Umsetzung von seinem Prioritätswert (4) in einen entsprechenden Dringlichkeitswert durch die Prioritätswert-Umsetzungseinrichtung 144 auf (Verbindung 200). In Fig. 5 liefert die Prioritätswert-Umsetzungseinrichtung 144, da der Prioritätswert (PRI) 4 im Computer 100 dem Dringlichkeitswert (UR) 1 entspricht, als Umsetzergebnis den Dringlichkeitswert 1 an das Benutzerprogramm 133 zurück (Verbindung 201). Dann fordert das Benutzerprogramm 133 das Benutzerprogramm 136 im Computer 101 dazu auf, eine Verarbeitung unter Verwendung des empfangenen Dringlichkeitswerts 1 gemäß der Dringlichkeitswert-Spezifizierform 190 oder 191 auszuführen (Verbindung 202). Der Computer 101 muß, wenn er die Information zum Spezifizieren des Dringlichkeitswerts 1 empfängt, diesen Dringlichkeitswert 1 in sein örtliches Prioritätsschema umsetzen und das Benutzerprogramm 136 mit demjenigen Prioritätswert ausführen, der sich aus der Umsetzung ergibt. Das Benutzerprogramm 136 fordert die Prioritätswert-Änderungseinrichtung 147 dazu auf, den spezifizierten Dringlichkeitswert 1 in einen entsprechenden Prioritätswert umzusetzen (Verbindung 203). Die Prioritätswert-Änderungseinrichtung 147 fordert die Prioritätswert-Umsetzungseinrichtung 145 dazu auf, den Dringlichkeitswert 1 in einen entsprechenden Prioritätswert im Computer 101 umzusetzen (Verbindung 204). In Fig. 5 entspricht der Dringlichkeitswert 1 im Computer 101 dem Prioritätswert 1. Die Prioritätswert-Umsetzungseinrichtung 145 versorgt die Prioritäts-Änderungseinrichtung 147 mit dem Prioritätswert 1 als Umsetzergebnis, um darüber zu informieren, daß das Benutzerprogramm 136 mit dem Prioritätswert 1 ausgeführt werden muß (Verbindung 205). Schließlich verlegt die Prioritätswert-Änderungseinrichtung 147 eine Programmstruktur 166 (entsprechend dem Benutzerprogramm 136), das nun mit einer Schlange für den Prioritätswert 4 verbunden ist, auf eine Schlange für den Prioritätswert 1 innerhalb der Prioritätsschlange 141 (Verbindung 206).

Auf diese Weise können die örtlichen Prioritätswerte, wie sie einem anfordernden Programm und einem angeforderten Programm, die in verschiedenen Computern liegen, zugehörige sind, gleichmäßig mittels des Dringlichkeitswerts verwaltet werden, der das gemeinsame Konzept im gesamten verteilten Computersystem ist. Wenn das eine Verarbeitung anfordernde Programm vorab seinen eigenen Dringlichkeitswert erkannt hat, ist keine Umsetzung vom örtlichen Prioritätswert im Computer auf den gemeinsamen Dringlichkeitswert erforderlich (Verbindungen 200, 201). Es ist auch möglich, eine Verarbeitungsanforderung mit einem Dringlichkeitswert auszugeben, der sich vom eigenen Dringlichkeitswert unterscheidet, und zwar einfach dadurch, daß direkt ein gewünschter Dringlichkeitswert spezifiziert wird.

Während die in Fig. 1 dargestellten Prioritätswert-Übersetzungstabellen 150, 151 jeweils Zuordnungen angesichts des Funktionsvermögens der zugehörigen



Computer 100 oder 101 vornehmen, hängt die Zeit für eine vollständige Verarbeitung tatsächlich von der Auslastung jedes Computers ab. Daher modifiziert dieses Ausführungsbeispiel die Prioritätswert-Entsprechungstabellen 150, 151 jedesmal dann, wenn die Auslastung eines zugehörigen Computers verändert wird. Fig. 6 zeigt einen Verarbeitungsablauf 210 zum Verändern des Inhalts einer Prioritätswert-Entsprechungstabelle.

Bei diesem Ausführungsbeispiel wird der Inhalt einer Prioritätswert-Entsprechungstabelle modifiziert, wenn die Auslastung eines Computers (die Anzahl ausführbarer Programme) um einen vorgegebenen Wert erhöht oder verringert wird. Der Unterschied der Auslastung gemäß der vorigen Prioritätswert-Entsprechungstabelle und gemäß der modifizierten Prioritätswert-Entsprechungstabelle wird als "gesamte Auslastungsänderung" bezeichnet. Wenn sich die Auslastung ändert, wird die aktuelle Auslastungsänderung zur Gesamtauslastungsänderung addiert (Schritt 211). Dann wird ermittelt, ob die sich ergebende Gesamtauslastungsänderung größer als ein vorgegebener positiver Wert (Schritt 212) oder kleiner als ein vorgegebener negativer Wert (Schritt 213) wird. Wenn die Gesamtauslastungsänderung größer als der vorgegebene positive Wert ist, bedeutet dies, daß die Auslastung um den vorgegebenen Wert zugenommen hat, so daß eine Auslastungsgrenze in der Prioritätswert-Entsprechungstabelle nach oben verschoben wird (Schritt 214). Umgekehrt bedeutet es, wenn die Gesamtauslastungsänderung kleiner als der negative vorgegebene Wert ist, daß die Auslastung um den vorgegebenen Wert abgenommen hat, wodurch die Dringlichkeitsgrenze in der Prioritätswert-Entsprechungstabelle nach unten verschoben wird (Schritt 215). Wenn die Inhalte der Prioritätswert-Entsprechungstabellen verändert werden, wird die Gesamtauslastungsänderung in einem abschließenden Schritt auf null gesetzt (Schritt 216). Dann wird die Gesamtauslastungsänderung erneut ausgehend von null aufsummiert.

Die Bedeutung, wie sie dem Verschieben der Dringlichkeitsgrenze in der Prioritätswert-Entsprechungstabelle nach oben oder unten im Schritt 214 bzw. im Schritt 213 innewohnt, wird unter Bezugnahme auf Fig. 7 erläutert. Wenn ein Anstieg der Auslastung eine Verschiebung der Dringlichkeitsgrenze in der Prioritätswert-Entsprechungstabelle 150 nach oben hervorruft, wird eine modifizierte Prioritätswert-Entsprechungstabelle 220 hergeleitet. Aus der Tabelle 220 ist es erkennbar, daß einem Programm mit demselben Dringlichkeitswert ein höherer Prioritätswert zugeordnet werden kann, als es in der Prioritätswert-Entsprechungstabelle 150 vor der Veränderung der Fall war. Umgekehrt wird dann, wenn eine Verringerung der Auslastung eine Verschiebung der Dringlichkeitsgrenze nach unten hervorruft, eine modifizierte Prioritätswert-Entsprechungstabelle 221 hergeleitet, wodurch einem Programm mit demselben Dringlichkeitswert ein niedrigerer Prioritätswert zugeordnet wird. Dadurch, daß die Dringlichkeitsgrenze auf diese Weise nach oben oder unten verschoben wird, kann selbst dann, wenn ein Programm mit demselben Dringlichkeitswert durch Computer mit demselben Funktionsvermögen (oder vom selben Computer) zu verarbeiten ist, das Programm mit höherem Prioritätswert ausgeführt werden, wenn der Computer stark ausgelastet ist, und mit niedrigerem Prioritätswert, wenn der Computer schwach ausgelastet ist. Auf diese Weise kann eine gleichmäßige Prioritätsverwaltung erzielt werden, wobei ferner Auslastungsänderungen in jedem Computer im verteilten Computer-

system berücksichtigt werden.

Während bei diesem Ausführungsbeispiel die Dringlichkeitsgrenze um einen Prioritätswert nach oben oder unten verschoben wird, kann das Ausmaß der Verschiebung abhängig vom Ausmaß der Auslastungsänderung verändert werden.

Nachfolgend wird ein Ablauf zum Verschieben der Dringlichkeitsgrenze nach oben oder unten erläutert, wenn die Prioritätswert-Entsprechungstabelle verwendet wird, wie sie durch die in Fig. 2 dargestellten Datenstrukturen 170 und 171 definiert ist. Fig. 8 zeigt Beispiele für den Ablauf, geschrieben in der Sprache C. Ein Block 230 repräsentiert den Ablauf zum Verschieben der Dringlichkeitsgrenze nach oben. Da "to\_riority [u]end" und "to\_riority [u+1]start" in der Datenstruktur 171 die untere Dringlichkeitsgrenze für den Dringlichkeitswert u angeben, werden diese Parameter verringert, um die Dringlichkeitsgrenze nach oben zu verschieben (Aussagen 240, 241). Die Prioritätswert-Entsprechungstabelle beinhaltet einen Bereich, in dem sich der Dringlichkeitswert, wie vom Prioritätswert aus gesehen, wegen der Verschiebung der Dringlichkeitsgrenze ändert. Z.B. ändert sich in Fig. 7 der dem Prioritätswert 1 entsprechende Dringlichkeitswert von 0 auf 1, und zwar wegen der Verschiebung der Dringlichkeitsgrenze nach oben. Aus diesem Grund muß der Inhalt des Felds 170 zum Umsetzen des Prioritätswerts in den Dringlichkeitswert abhängig von der Änderung des Dringlichkeitswerts modifiziert werden. Wenn die Dringlichkeitsgrenze nach oben geschoben wird, ist "to\_riority [u+1]start" der Bereich, in dem sich der Dringlichkeitswerts ändert, wie er vom Prioritätswert aus gesehen wird. Daher wird der Wert von "to\_urgency [to\_riority [u+1]. start]" zur Korrektur erhöht (Aussage 242).

Es ist hier zu beachten, daß eine Änderung der Prioritätswert-Entsprechungstabelle zu einer Änderung des Dringlichkeitswerts führen kann, wie er für ein Programm eingestellt ist, das bereits in der Prioritätsschlange vorliegt. Z.B. bewirkt in Fig. 7 ein nach oben Verschieben der Dringlichkeitsgrenze eine Änderung des Dringlichkeitswerts eines Programms mit dem Dringlichkeitswert 0 und dem Prioritätswert 1 von 0 auf 1. Im Ergebnis hat dieses Programm den Dringlichkeitswert 1 und den Prioritätswert 1 als Attribute. Bei diesem Ausführungsbeispiel kann, da dieses Programm so korrigiert wird, daß es den Dringlichkeitswert 0 aufweist, ein in der später beschriebenen Fig. 9 dargestellter Ablauf innerhalb der Prioritätswert-Änderungseinrichtungen 146, 147 verwendet werden. Der Ablauf von Fig. 9 arbeitet so, daß er den Prioritätswert eines Programms ändert, dessen Dringlichkeitswert aufgrund einer Änderung der Prioritätswert-Entsprechungstabelle geändert wird, so daß das Programm denselben Dringlichkeitswert beibehält, den es vor dem Ändern der Prioritätswert-Entsprechungstabelle hatte. Bei oben genannten Beispiel wird dann, wenn der Prioritätswert aller Programme mit dem Prioritätswert 1 um 1 verringert wird, so daß diese Programme den Prioritätswert 0 haben, der Dringlichkeitswert dieser Programme auf den Wert (Dringlichkeitswert 0) korrigiert, den sie vor der Änderung der Prioritätswert-Entsprechungstabelle hatten. Eine Aussage 243 ist eine Funktion zum Aufrufen des in Fig. 9 dargestellten Verarbeitungsablaufs aus den Prioritätswert-Umsetzungseinrichtungen 144, 145. In der Aussage 243 ist angenommen, daß der Name der in Fig. 9 dargestellten Funktion "change\_all\_priority" ist und daß zu spezifizierende Argumente "to\_riority [u+1]start"



sind, wobei es sich um einen Bereich handelt, in dem sich die Entsprechung zwischen dem Dringlichkeitswert und dem Prioritätswert geändert hat, mit einem Anstieg des Prioritätswerts (es ist zu beachten, daß in einer Aussage 243 das Inkrement des Prioritätswerts  $-1$  ist, so daß eine Verringerung des Prioritätswerts vorgegeben ist).

Fig. 9 zeigt unter den Funktionen, die in den Prioritätswert-Änderungseinrichtungen 146, 147 vorliegen, einen Verarbeitungsablauf 250 zum Ändern des Prioritätswerts aller Programme mit einem speziellen Prioritätswert auf einen anderen Wert. Zunächst ruft der Ablauf 250 eine einem speziellen Prioritätswert entsprechende Schlange aus der Prioritätsschlange 140 oder 141 ab (Schritt 251) und untersucht dann, ob die Schlange leer ist oder nicht (Schritt 252). Wenn die Schlange nicht leer ist, wird die Programmstruktur in der Schlange abgerufen (Schritt 253) und der Prioritätswert der Programme wird um einen vorgegebenen Wert inkrementiert (Schritt 254). Wenn der angegebene Wert negativ ist, wird der Prioritätswert verringert. Es ist zu beachten, daß bei diesem Beispiel ein verringerter Prioritätswert einen höheren Prioritätsrang bedeutet. Dann wird die abgerufene Programmstruktur an eine Schlange verlegt, die dem Prioritätswert entspricht, der um den vorgegebenen Wert erhöht ist (Schritt 255). Wenn ein negativer Wert addiert wird, verringert sich der entsprechende Prioritätswert. So wird die von der Aussage 243 angeforderte Prioritätswertänderung ausgeführt, wodurch der Prioritätswert von Programmen so geändert werden kann, daß derselbe Dringlichkeitswert aufrechterhalten bleibt.

Ähnlich wie im vorstehend genannten Fall repräsentiert ein Block 231 in Fig. 8 einen Ablauf zum Verschieben der Dringlichkeitsgrenze nach unten. Diese Verschiebung nach unten wird dadurch erhöht, daß "to\_prio-  
[u].end" und "to\_priority[u+1].start" in der Datenstruktur 171 erhöht werden (Aussagen 244, 245). Wenn die Dringlichkeitsgrenze nach unten verschoben wird, ist ein Bereich, in dem der Dringlichkeitswert sich so ändert, wie vom Prioritätswert aus gesehen, "to\_prio-  
[u].end". Daher wird der Wert von "to\_urgency[to\_prio-  
[u].end]" zur Korrektur herabgesetzt (Aussage 246). Der Prioritätswert von Programmen kann durch eine Aussage 247 geändert werden, wie im Fall des Ablaufs 230. In diesem Fall werden "to\_priority[u].end", wobei es sich um einen Bereich handelt, in dem sich die Entsprechung zwischen dem Dringlichkeitswert und dem Prioritätswert geändert hat, und das Inkrement für den Prioritätswert ( $= 1$ ) als Argumente spezifiziert.

Unter Bezugnahme auf die Fig. 1—9 wurden die Konfiguration, der Betrieb und Abläufe beim ersten Ausführungsbeispiel beschrieben. Das erste Ausführungsbeispiel verwendet das Konzept der Dringlichkeit, um es einem verteilten Computersystem zu ermöglichen, örtliche Prioritätsschemata mehrerer Computer gleichmäßig zu verwalten, die sich hinsichtlich des Funktionsvermögens, der Auslastung und des Typs voneinander unterscheiden. Wenn der Dringlichkeitswert für diese Computer vorgegeben wird, ist es zu erwarten, daß eine angeforderte Verarbeitung mit im wesentlichen derselben Geschwindigkeit in jedem dieser Computer abgeschlossen wird. Auch ist es möglich, da der Dringlichkeitswert unter Berücksichtigung des Funktionsvermögens und der Auslastung von Computern in einen Prioritätswert umgesetzt wird, einen Umkehr Effekt zu vermeiden, bei dem eine Verarbeitung mit niedrigerem Dringlichkeitswert vor einer Verarbeitung mit höherem Dringlichkeitswert abgeschlossen wird, und zwar selbst

dann, wenn sie auf verschiedenen Computern abgearbeitet werden. Zusammengefaßt gesagt, gewährleistet das erste Ausführungsbeispiel dank des Konzepts der Dringlichkeit eine strengere Einhaltung von Echtzeiteigenschaften in einem verteilten Computersystem.

Ferner kann bei diesem Ausführungsbeispiel, da die Inhalte der Prioritätswert-Entsprechungstabellen und der Prioritätsschlangen modifiziert werden können (wie in Verbindung mit den Fig. 6—9) beschrieben, das Prioritätsschema abhängig von sich dynamisch ändernden Auslastungen jeweiliger Computer verwaltet werden.

Ferner kann zusätzlich zu den oben genannten zwei Haupteffekten dann, wenn eine Verarbeitung bei zwei oder mehr Computern angefordert wird, der Dringlichkeitswert für die Verarbeitung abhängig von der Wichtigkeit spezifiziert werden. Außerdem ist es nicht erforderlich, das Funktionsvermögen, die Auslastung und den Typ zusammenwirkender Computer zu kennen, um den Dringlichkeitswert für eine Verarbeitung anzugeben.

Während beim ersten Ausführungsbeispiel die Auslastung eines Computers durch die Anzahl ausführbarer Programme repräsentiert ist, kann die Auslastung durch die Summe der Verarbeitungsmengen ausführbarer Programme und die vorausgesagte Verarbeitungszeit repräsentiert sein. Der letztgenannte Weg führt zu genaueren Auslastungsberechnungen für die jeweiligen Computer, wodurch strenger eingehaltene Echtzeiteigenschaften gewährleistet werden. Ferner kann, während der Prioritätswert unter Berücksichtigung sowohl der Auslastung als auch des Funktionsvermögens jedes Computers bestimmt wird, die Bestimmung nur gestützt auf eine dieser Angaben erfolgen. Insbesondere dann, wenn ein verteiltes Computersystem aus jeweils denselben Computern besteht, muß nur die Auslastung berücksichtigt werden. Auf diese Weise kann die Umsetzung zwischen Dringlichkeitswerten und Prioritätswerten auf wirkungsvolle Weise ausgeführt werden.

Nachfolgend wird die Konfiguration eines verteilten Computersystems gemäß einem zweiten Ausführungsbeispiel der Erfindung unter Bezugnahme auf Fig. 10 erläutert. Das in Fig. 10 dargestellte zweite Ausführungsbeispiel zeichnet sich durch eine Prioritätswert-Umsetzeinrichtung 260 zum Verwalten der Prioritätswerte aller Computer in einem verteilten Computersystem aus. Die Elemente 100, 101, 110, 130—136, 140—143, 146, 147 sowie 160—166 sind mit denen identisch, die in Fig. 1 mit denselben Bezugszahlen bezeichnet sind. In Fig. 10 sind Veranschaulichungen der Computer 102, 103 und Einzelheiten innerhalb der Computer weggelassen. Die Prioritätswert-Umsetzeinrichtung 260 verfügt über eine Prioritätswert-Entsprechungstabelle 270 zum Angeben der Entsprechung zwischen Prioritätswerten und Dringlichkeitswerten für alle Computer. Die Datenstruktur der Prioritätswert-Entsprechungstabelle 270 ist dergestalt, daß die in Fig. 2 dargestellten Datenstrukturen 170, 171 für die Anzahl von Computern im verteilten Computersystem vorbereitet sind. Daher werden Modifizierungen an der Prioritätswert-Entsprechungstabelle 270 und entsprechende Änderungen der Prioritätswerte von Programmen mit denselben Abläufen ausgeführt, wie sie unter Bezugnahme auf die Fig. 6—9 erläutert wurden.

Fig. 10 beinhaltet auch eine Veranschaulichung eines Konzepts eines Verarbeitungsablaufs zum Anfordern einer Verarbeitung mit spezifiziertem Dringlichkeitswert. Bei diesem Ausführungsbeispiel ist angenommen, daß der Verarbeitungsablauf Schritte repräsentiert, wie sie ausgeführt werden, wenn ein Benutzerprogramm



133 im Computer 100 eine Verarbeitungsanforderung an ein Benutzerprogramm 136 im Computer 101 ausgibt, wie es bei Fig. 5 der Fall ist. Zunächst fordert das Benutzerprogramm 133 die Prioritätswert-Umsetzeinrichtung 260 dazu auf, den Prioritätswert in einen entsprechenden Dringlichkeitswert umzusetzen (Verbindung 280). Wenn das Benutzerprogramm 133 und die Prioritätswert-Umsetzeinrichtung 260 in voneinander verschiedenen Computern vorhanden sind, ist die Verbindung 280 eine solche zwischen verschiedenen Computern. Da die Prioritätswert-Entsprechungstabelle 270 anzeigt, daß der Prioritätswert 4 im Computer 100 dem Dringlichkeitswert 1 entspricht, liefert die Prioritätswert-Umsetzeinrichtung 260 den Dringlichkeitswert 1 an das Benutzerprogramm 133 zurück (Verbindung 281). Das Benutzerprogramm 133 gibt als nächstes eine Verarbeitungsanforderung an das Benutzerprogramm 136 aus, wobei es den empfangenen Dringlichkeitswert 1 verwendet (Verbindung 282). Der Computer 101 setzt dann, wenn er die Information zum Spezifizieren des Dringlichkeitswerts 1 empfängt, diesen Dringlichkeitswert 1 im Computer 101 in einen örtlichen Prioritätswert um, und er weist das Benutzerprogramm 136 an, die angeforderte Verarbeitung mit dem hergeleiteten Prioritätswert auszuführen. Das Benutzerprogramm 136 fordert die Prioritätswert-Änderungseinrichtung 147 dazu auf, den spezifizierten Dringlichkeitswert 1 in einen entsprechenden Prioritätswert zu ändern (Verbindung 283). Die Prioritätswert-Änderungseinrichtung 147 fordert die Prioritätswert-Umsetzeinrichtung 260 dazu auf, den spezifizierten Dringlichkeitswert 1 in einen entsprechenden Prioritätswert umzusetzen (Verbindung 284). Bei diesem Ausführungsbeispiel entspricht der Dringlichkeitswert 1 im Computer 101 dem Prioritätswert 1. Die Prioritätswert-Umsetzeinrichtung 260 liefert den Prioritätswert 1 als Umsetzungsergebnis an die Prioritätswert-Änderungseinrichtung 147 zurück, um darüber zu informieren, daß das Benutzerprogramm 136 mit dem Prioritätswert 1 ausgeführt werden muß (Verbindung 285). Da das Benutzerprogramm 136 und die Prioritätswert-Umsetzeinrichtung 260 in voneinander verschiedenen Computern liegen, sind die Verbindungen 284, 285 solche zwischen verschiedenen Computern. Schließlich verlegt die Prioritätswert-Änderungseinrichtung 147 eine Programmstruktur 166 in einer Prioritätsschlange 141 in eine Schlange für den Prioritätswert 1 (Verbindung 286).

Es ist zu beachten, daß dann, wenn das Verarbeitungsanforderungsprogramm zuvor seinen eigenen Dringlichkeitswert erkannt hat, eine Umsetzung vom örtlichen Prioritätswert im Computer, in dem das Programm liegt, auf einen gemeinsamen Dringlichkeitswert (Verbindungen 280, 281) nicht erforderlich ist, wie im Fall von Fig. 5. Es ist auch möglich, daß ein Programm eine Verarbeitungsanforderung mit einem Dringlichkeitswert ausgibt, der sich vom eigenen Dringlichkeitswert unterscheidet, und zwar einfach durch direktes Spezifizieren eines speziellen Dringlichkeitswerts.

Das vorstehend erläuterte zweite Ausführungsbeispiel entspricht der Prioritätswert-Umsetzeinrichtung beim ersten Ausführungsbeispiel, wie es in Verbindung mit den Fig. 1—9 beschrieben wurde, mit einer gemeinsamen Anordnung in einem einzelnen Computer, weswegen der Vorteil besteht, daß die Prioritätswert-Entsprechungstabelle extrem einfach verwaltet werden kann, zusätzlich zu den Wirkungen, die beim ersten Ausführungsbeispiel vorliegen. Die Konfiguration des zweiten Ausführungsbeispiels ist für ein Konzept geeignet,

bei dem ein Computer mit hohem Funktionsvermögen und hoher Zuverlässigkeit in einem verteilten Computersystem vorhanden ist, um das gesamte System zu verwalten, so daß die Verarbeitungseinrichtungen, die mit der Prioritätswertverwaltung in Beziehung stehen, gemeinsam in diesem Computer liegen.

Nachfolgend wird ein drittes Ausführungsbeispiel der Erfindung beschrieben. Innerhalb eines verteilten Computersystems ist die Aufteilung derselben Menge an Programmen auf jeweilige Computer zum gleichen Auslasten derselben auch von Nutzen, um Echtzeiteigenschaften sicherzustellen. Wenn jedoch die gleiche Anzahl von Programmen auf alle zur Verfügung stehenden Computer verteilt wird, kann nicht gesagt werden, daß die Computermittel wirkungsvoll genutzt sind, und zwar wenn eilige Programme in einem Computer konzentriert sind und einfache Programme als Ergebnis der Auslastungsverteilung in einem anderen Computer konzentriert sind. Auch führt die Konzentration eiliger Programme in einem Computer zu einer Verzögerung der Verarbeitung im Computer, wodurch keine Echtzeiteigenschaften sichergestellt werden können. Die Erfindung erzielt jedoch eine Auslastungsverteilung abhängig von der Wichtigkeit jedes Programms, und zwar unter Verwendung der Dringlichkeit. Fig. 1 zeigt den Aufbau eines verteilten Computersystems gemäß einem dritten Ausführungsbeispiel zum Ausführen einer solchen Auslastungsverteilung. Die Elemente 100—103, 110, 130—136, 140—143, 146, 147 sowie 160—166 in Fig. 11 sind mit denjenigen Elementen identisch, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind, während eine Prioritätswert-Umsetzeinrichtung 260 und eine Prioritätswert-Entsprechungstabelle 270 mit den in Fig. 10 dargestellten identisch sind. Die Betriebsabläufe in diesen Elementen in Fig. 11 sind ebenfalls ähnlich zu denen der Gegenstücke in den Fig. 1 und 10. Ein Benutzerprogramm 137 wie auch Benutzerprogramme 130—136 liegen in der Hauptspeichereinheit eines Computers 101, so daß das Benutzerprogramm 137 durch eine Programmstruktur 167 in einer Prioritätsschlange 140 verwaltet wird.

Programmausführungs- und Verschiebeeinrichtungen 290, 291, die dazu dienen, ein Programm zu starten bzw. ein Programm vom zugehörigen Computer zu einem anderen zu verschieben, sind normalerweise als Funktionen der Betriebssysteme der jeweiligen Computer vorhanden. Der Start eines Programms beinhaltet das Erzeugen einer Programmstruktur für das neu auszuführende Programm und das Einschreiben der Programmstruktur in die Prioritätsschlange 140 oder 141, wodurch das Programm in einen ausführbaren Zustand gebracht wird. Das Verschieben eines Programms beinhaltet das Verlegen eines Programms in einen ausführbaren Zustand in einem Computer in einen anderen Computer und das Überführen des verlegten Programms in ausführbaren Zustand im Zielcomputer. Diese Funktion wird auch als "Prozeßverlegung" bezeichnet, die häufig als Funktion eines verteilten Betriebssystems vorhanden ist.

Ein Computerinformationsbereich 300 besteht aus einer Computerauslastungs-Informationstabelle 302 und einer Computerfunktionsvermögen-Informationstabelle 304. Die Computerauslastungs-Informationstabelle 302 beinhaltet Auslastungswerte für jeweilige Computer, d. h. die Anzahl von Programmen, wie sie von den jeweiligen Computern für jeden Dringlichkeitswert auszuführen sind. Z.B. ist es aus Fig. 11 erkennbar, daß gemäß der aktuellen Auslastung der Computer 100 drei





Programme mit dem Dringlichkeitswert 0, acht Programme mit dem Dringlichkeitswert 1, sieben Programme mit dem Dringlichkeitswert 2 usw. beinhaltet, während gemäß der aktuellen Auslastung der Computer 101 ein Programm mit dem Dringlichkeitswert 0, ein Programm mit dem Dringlichkeitswert 1, zwei Programme mit dem Dringlichkeitswert 2 usw. beinhaltet. Die Computerfunktionsvermögen-Informationstabelle 304 speichert Angaben zu den Funktionsvermögen der jeweiligen Computer. Bei diesem Ausführungsbeispiel ist das Funktionsvermögen durch einen Zahlenwert gekennzeichnet, wobei z. B. 10 MIPS dem Wert 1 entsprechen. Das Funktionsvermögen des Computers 100 ist 10, und das Funktionsvermögen des Computers 101 ist gemäß dieser Wertezuordnung 3. Die Programmausführungseinrichtung 209 und die Programmverschiebeeinrichtung 291 geben nach dem Starten oder Verschieben eines Programms Änderungen wieder, wie sie wegen ihrer Abläufe möglicherweise für die Computerauslastung-Informationstabelle 302 vorliegen. Dadurch kann die Computerauslastung-Informationstabelle 302 jederzeit den Auslastungszustand des gesamten verteilten Computersystems korrekt angeben.

Eine Programmverschiebung-Bestimmungseinrichtung 292 bestimmt aus der Computerauslastung-Informationstabelle 302 im Computerinformationsbereich 300, ob die jeweiligen Computer abhängig von ihren Funktionsvermögen gleich ausgelastet sind. Wenn die Auslastung nicht gleichmäßig verteilt ist, bestimmt die Einrichtung 292, ein Programm zu verschieben, um gleiche Verteilung der Auslastung zu erzielen. Bei diesem Ausführungsbeispiel ist angenommen, daß die Programmverschiebe-Bestimmungseinrichtung 292 periodisch, also mit regelmäßigen Intervallen, gestartet wird. Eine Computerauswahleinrichtung 294 bestimmt, welcher Computer der Ausführung eines neu angeforderten Programms zugeordnet werden soll und welcher Computer als Ziel für ein zu verschiebendes Programm auszuwählen ist, was unter Bezugnahme auf die Computerauslastung-Informationstabelle 302 und die Computerfunktionsvermögen-Informationstabelle 304 im Computerinformationsbereich 300 erfolgt.

Der Betrieb dieses Ausführungsbeispiels wird unter Bezugnahme auf die Fig. 11–15 beschrieben. Zunächst wird der Vorgang erläutert, wenn ein Programm neu gestartet wird. Wenn ein Benutzer einen Befehl zum Spezifizieren eines Dateinamens eines ausführbaren Programms und dessen Dringlichkeitswert eingibt, analysiert das Betriebssystem den Befehl, und es fordert die Programmausführungseinrichtung 290 oder die Programmverschiebeeinrichtung 291 im Betriebssystem dazu auf, das Programm zu starten, wodurch das Starten des Programms ermöglicht ist. Hier wird der Fall erläutert, daß ein Befehl zum Starten eines Programms mit dem Dringlichkeitswert 1 in den Computer 100 eingegeben wird und die Programmausführungs- und Verschiebeeinrichtung 290 darauf ansprechend einen Vorgang ausführt, wie er dem Start eines Programms zugeordnet ist.

Unter Bezugnahme auf das Flußdiagramm der Fig. 12 wird ein Verarbeitungsablauf 310 zum Starten eines Programms erläutert. Zunächst fordert die Programmausführungs- und Verschiebeeinrichtung 290 die Computerauswahleinrichtung 294 dazu auf, einen Auslastung/Funktionsvermögen-Wert im Bereich des Dringlichkeitswerts 0 bis zum Wert auszurechnen, wie er für das zu startende Programm vorgegeben ist, und zwar für alle Computer im verteilten Computersystem

(Schritt 311). Der Auslastung/Funktionsvermögen-Wert wird dadurch berechnet, daß der Wert für die Auslastung eines Computers durch den Wert für dessen Funktionsvermögen geteilt wird. Bei diesem Ausführungsbeispiel wird die Anzahl der in der Prioritätsschlange in einem Computer eingetragenen ausführbaren Programme als Auslastungswert für den Computer verwendet. Die Berechnung wird dadurch ausgeführt, daß die in der Computerauslastung-Informationstabelle 302 und der Computerfunktionsvermögen-Informationstabelle 304 im Computerinformationsbereich 300 abgespeicherten Werte ausgelesen werden. Fig. 13 zeigt die Auslastungsinformation, die Auslastungswerte und die Auslastung/Funktionsvermögen-Werte für die jeweiligen Computer, die sich in dem in Fig. 11 dargestellten Zustand befinden. Die Computerauslastung-Informationstabelle 302 gibt die Anzahl ausführbarer Programme für jeden Dringlichkeitswert an. Die Auslastungswerttabelle 306 gibt die Anzahl ausführbarer Programme in jedem Bereich vom Dringlichkeitswert 0 bis zu einem speziellen Dringlichkeitswert (1, 2, 3, ... 31) an, und die Auslastung/Funktionsvermögen-Wertetabelle 308 speichert Auslastung/Funktionsvermögen-Werte ab, die dadurch gewonnen wurden, daß die Auslastungswerte durch die jeweiligen Werte für das Computerfunktionsvermögen geteilt wurden. Da die Erörterung nun für einen Fall abläuft, bei dem ein neu zu startendes Programm den Dringlichkeitswert 1 hat, wird der Auslastung/Funktionsvermögen-Wert für den Bereich 0–1 der Dringlichkeitswerte zu 1,10 für den Computer 100 und zu 0,67 für den Computer 101 berechnet.

Es wird erneut auf Fig. 12 Bezug genommen, gemäß der die Computerauswahleinrichtung 294 als nächstes denjenigen Computer mit dem kleinsten Auslastung/Funktionsvermögen-Wert als Computer auswählt, der der Ausführung des zu startenden Programms zugeordnet wird (Schritt 312). Obwohl in Fig. 13 die Auslastung/Funktionsvermögen-Werte für mit 102 und höheren Bezugswerten gekennzeichnete Computer weggelassen ist, sei angenommen, daß der Auslastung/Funktionsvermögen-Wert für den Computer 101, der 0,67 ist, der kleinste Wert ist, weswegen dieser Computer 101 ausgewählt wird. Schließlich führt die Programmausführungs- und Verschiebeeinrichtung 290 einen Vorgang aus, der den ausgewählten Computer dazu anweist, das Programm zu starten (Schritt 313). Dieser Vorgang erfolgt auf solche Weise, daß die Programmausführungs- und Verschiebeeinrichtung des Computers, der mit dem Programmstart begonnen hat, die Programmausführungs- und Verschiebeeinrichtung des ausgewählten Computers dazu auffordert, das Programm in ausführbaren Zustand zu bringen. Bei diesem Beispiel fordert die Programmausführungs- und Verschiebeeinrichtung 291 des Computers 100 die Programmausführungs- und Verschiebeeinrichtung 191 im Computer 101 dazu auf, das Programm in ausführbaren Zustand zu bringen.

Nachfolgend erfolgt eine Erläuterung hinsichtlich des Betriebs zum Verschieben eines ausführbaren Programms. Bei diesem Ausführungsbeispiel wird die Programmverschiebe-Bestimmungseinrichtung 292 periodisch gestartet, um zu überprüfen, ob die Auslastung gleichmäßig auf die jeweiligen Computer abhängig von ihren Funktionsvermögen verteilt ist. Wenn die Verteilung nicht gleichmäßig erfolgt ist, bestimmt die Programmverschiebe-Bestimmungseinrichtung 292 die Verschiebung eines Programms, um zu einer gleichmäßigen Verteilung der Auslastung zu kommen, und dann fordert sie die Programmausführungs- und Verschiebe-



einrichtung 290 dazu auf, ein Programm zu verschieben. Die Programmausführungs- und Verschiebeeinrichtung 290 fordert ihrerseits die Computerauswahleinrichtung 294 dazu auf, einen Zielcomputer auszuwählen, und sie führt einen Vorgang aus, der der Verschiebung eines Programms zugeordnet ist, und zwar in Zusammenwirkung mit der Programmausführungs- und Verschiebeeinrichtung des ausgewählten Zielcomputers. Bei diesem Ausführungsbeispiel wird der oben genannte Auslastung/Funktionsvermögen-Wert dazu verwendet, zu bestimmen, ob die Auslastungen der jeweiligen Computer unter Berücksichtigung ihrer Funktionsvermögen gleich verteilt sind.

Unter Bezugnahme auf das Flußdiagramm von Fig. 14 wird ein Verarbeitungsablauf 320 zum Verschieben eines Programms erläutert. Zunächst setzt die Programmverschiebe-Bestimmungseinrichtung 292 eine Variable  $n$ , die den Bereich für den Dringlichkeitswert kennzeichnet, auf den Wert 0 (Schritt 321). Dann werden die Auslastung/Funktionsvermögen-Werte im Bereich der Dringlichkeitswerte  $0-n$  für die jeweiligen Computer berechnet (Schritt 322). Wie oben beschrieben, sind die Auslastung/Funktionsvermögen-Werte, wie sie für den in Fig. 11 dargestellten Zustand berechnet werden, diejenigen, die als Auslastung/Funktionsvermögen-Werte in der Tabelle 308 in Fig. 11 dargestellt sind. Danach wird die Differenz  $D$  zwischen dem größten Auslastung/Funktionsvermögen-Wert und dem kleinsten Auslastung/Funktionsvermögen-Wert für alle Computer innerhalb des spezifizierten Bereichs der Dringlichkeitswerte berechnet (Verarbeitung 323). Obwohl die Auslastung/Funktionsvermögen-Werte für die mit der Bezugszahl 102 und höheren Bezugszahlen gekennzeichneten Computer in Fig. 13 weggelassen sind, wird angenommen, daß der Last/Funktionsvermögen-Wert für den Computer 101, der 0,33 ist, der größte ist, und daß der Last/Funktionsvermögen-Wert für den Computer 100, der 0,30 ist, der kleinste ist, wenn  $n$  den Wert 0 hat, und demgemäß wird die Differenz  $D$  zu 0,03 berechnet. Dann wird ermittelt, ob dieser Differenzwert  $D$  einen Schwellenwert  $D_t$  überschreitet (Schritt 324). Bei diesem Ausführungsbeispiel ist angenommen, daß der Schwellenwert  $D_t$  auf 0,3 gesetzt ist. Wenn der Differenzwert  $D$  den Schwellenwert  $D_t$  übersteigt, geht der Ablauf zu einem Schritt 325 weiter, andernfalls zu einem Schritt 328. Da bei diesem Beispiel die Differenz  $D$  den Schwellenwert  $D_t$  nicht übersteigt, wenn  $n$  null ist, geht der Ablauf zum Schritt 328 weiter, in dem  $n$  um eins inkrementiert wird. Danach wird ermittelt, ob  $n$  den Wert 31 übersteigt (Schritt 329). Der Ablauf wird beendet, wenn  $n$  den Wert 31 überschreitet, oder er springt auf den Schritt 322 zurück, wenn dies nicht der Fall ist.

Da  $n$  nun auf eins gesetzt ist, kehrt der Ablauf zum Schritt 322 zurück, in dem die Auslastung/Funktionsvermögen-Werte für die jeweiligen Computer berechnet werden, und dann wird wieder der Differenzwert  $D$  im Schritt 323 berechnet. Obwohl die Auslastung/Funktionsvermögen-Werte der mit 102 und den anschließenden Bezugszahlen gekennzeichneten Computer in Fig. 13 weggelassen sind, ist angenommen, daß der Auslastung/Funktionsvermögen-Wert des Computers 100 mit 1,10 der größte Wert ist und der Auslastung/Funktionsvermögen-Wert des Computers 101 von 0,67 der kleinste ist, wenn  $n$  den Wert eins hat, weswegen die Differenz  $D$  zu 0,34 berechnet wird. Daher überschreitet die Differenz  $D$  den Schwellenwert  $D_t$ , wenn  $n$  den Wert eins hat. Wenn die Differenz  $D$  den Schwellenwert  $D_t$  überschreitet, entspricht dies der Ermittlung, daß die

Auslastung nicht gleichmäßig verteilt ist, so daß ein zu verschiebendes Programm bestimmt wird (Schritt 325).

Bei diesem Ausführungsbeispiel wird das als letztes auszuführende Programm, d. h. dasjenige, das in der Prioritätsschlange als letztes steht, aus den ausführbaren Programmen für den Dringlichkeitswert  $n$  im Computer mit dem größten Auslastung-Funktionsvermögen-Wert ausgewählt. Wie es in Fig. 11 dargestellt ist, wird dieses Programm K zur Verschiebung ausgewählt, da der dem Dringlichkeitswert 1 des Computers 100 entsprechende Prioritätswertbereich 2-4 ist und das als letztes auszuführende Programm das Programm K ist (Programmstruktur 167).

Wenn das zu verschiebende Programm bestimmt ist, fordert die Programmverschiebe-Bestimmungseinrichtung 292 die Programmausführungs- und Verschiebeeinrichtung des Computers, der das zu verschiebende Programm enthält, dazu auf, den Vorgang der Verschiebung des ausgewählten Programms auszuführen. Bei diesem Beispiel führt die Programmausführungs- und Verschiebeeinrichtung 290 des Computers 100 den Programmverschiebevorgang aus. Zunächst fordert die Programmausführungs- und Verschiebeeinrichtung 290 die Computerauswahleinrichtung 294 dazu auf, einen Zielcomputer auszuwählen, an den das Programm zu verschieben ist (Schritt 326 in Fig. 14). Ein als Ziel ausgewählter Computer ist bei diesem Ausführungsbeispiel derjenige Computer mit dem kleinsten Auslastung/Funktionsvermögen-Wert. Daher wird der Computer 101 mit dem kleinsten Auslastung/Funktionsvermögen-Wert, wenn  $n$  den Wert eins hat, als Ziel für das Programm ausgewählt.

Wenn das Ziel bestimmt ist, führt die Programmausführungs- und Verschiebeeinrichtung den Programmverschiebevorgang aus (Schritt 327). Bei diesem Ausführungsbeispiel führt die Programmausführungs- und Verschiebeeinrichtung 290 des Computers 100 den Programmverschiebevorgang in Zusammenarbeit mit der Programmausführungs- und Verschiebeeinrichtung 291 des Computers 101 aus, der das Ziel für das Programm K ist.

Normalerweise ist die Programmausführungs- und Verschiebeeinrichtung durch das Betriebssystem gebildet. Um ein Programm zu verschieben, ist der folgende Vorgang erforderlich, um das Programm ausführbar zu machen, wenn der Zielcomputer 101 über keine ausführbare Datei für das zu verschiebende Programm verfügt. Vom Computer 100 wird eine ausführbare Datei auf den Computer 101 verlegt, die Programmstruktur des verschobenen Programms K im Computer 100 wird aus der Prioritätsschlange entnommen, und im Zielcomputer 101 wird eine Programmstruktur erzeugt und zur Prioritätsschlange hinzugefügt. Nachdem das Programm verschoben ist, spiegelt sich der veränderte Auslastungszustand des Computers in der Computerauslastungs-Informationstabelle 302 wider. Bei diesem Ausführungsbeispiel wird die Computerauslastungs-Informationstabelle 302 modifiziert, wonach die Anzahl ausführbarer Programme mit dem Dringlichkeitswert 1 für den Computer 100 ist, dagegen zwei für den Computer 101, wie in Fig. 15 dargestellt.

Nachdem das Programm wie oben angegeben verschoben ist, kehrt der Ablauf zum Schritt 322 zurück, um erneut die Auslastung/Funktionsvermögen-Werte für die jeweiligen Computer zu berechnen. In diesem Fall werden die berechneten Auslastungswerte und Auslastung/Funktionsvermögen-Werte so geändert, wie es in den Tabellen 306 bzw. 308 in Fig. 15 dargestellt ist. Ob-





wohl in Fig. 15 die berechneten Werte für die mit 102 und anschließenden Bezugswerten gekennzeichneten Computer weggelassen sind, wird der Programmverschiebevorgang wie oben angegeben wiederholt, wenn der Differenzwert D immer noch den Schwellenwert Dt übersteigt, wenn n den Wert eins hat. Wenn der Differenzwert D so groß ist wie der Schwellenwert Dt, oder kleiner ist, geht der Ablauf zum Schritt 328 weiter.

Durch Wiederholen der vorstehend genannten Vorgänge, bis n den Wert 31 (den niedrigsten Dringlichkeitsrang) überschreitet, wird eine Anzahl von Programmen verschoben, um eine gleichmäßige Verteilung der Auslastung für alle Dringlichkeitswerte zu erzielen, d. h., um die Auslastung auf alle Computer abhängig von ihren Funktionsvermögen gleich zu verteilen.

Während die vorstehend genannte Programmverschiebung eine solche ist, die von der Programmverschiebe-Bestimmungseinrichtung für die Auslastungsverteilung gestartet wird, kann diese Programmverschiebung auch abhängig von einem vom Benutzer eingegebenen Befehl ausgeführt werden. In diesem Fall wird der Dringlichkeitswert für ein zu verschiebendes Programm auf n gesetzt und es werden die Schritte 326, 327 in Fig. 14 ausgeführt, wodurch das Programm auf den am geringsten ausgelasteten Computer verschoben werden kann.

Es wurde nun das verteilte Computersystem gemäß dem dritten Ausführungsbeispiel der Erfindung beschrieben. Dieses Ausführungsbeispiel ist dahingehend vorteilhaft, daß allen Computern abhängig von ihren Funktionsvermögen Benutzerprogramme in gleicher Weise zugeordnet werden können, um die Auslastung im gesamten verteilten Computersystem auf gleiche Weise zu verteilen, so daß die Computermittel wirkungsvoll genutzt werden. Da das System dieses Ausführungsbeispiel die Gleichverteilung der Auslastung unter Berücksichtigung sowohl der Auslastungswerte als auch der Funktionsvermögen der jeweiligen Computer bewertet, kann die Auslastungsverteilung selbst dann klar abgeschätzt werden, wenn Differenzen zwischen den Funktionsvermögen verschiedener Computer vorliegen.

Auch können bei diesem Ausführungsbeispiel Benutzerprogramme gleichmäßig Computern abhängig von deren jeweiligen Funktionsvermögen für jeden Dringlichkeitswert von Programmen zugeordnet werden, so daß Echtzeitverarbeitung mit gleichmäßiger Dringlichkeit im gesamten verteilten Computersystem aufrechterhalten werden kann.

Insbesondere ist es möglich, da die Auslastungsverteilung unter Berücksichtigung des Dringlichkeitswerts für jedes Programm ausgeführt wird, zu verhindern, daß dringlich auszuführende Programme auf einen Computer konzentriert werden, wodurch Echtzeiteigenschaften strenger eingehalten werden.

Ferner wird gemäß diesem Ausführungsbeispiel die Verwaltung der Auslastung im gesamten verteilten Computersystem durch einen einzelnen Computer auf konzentrierte Weise ausgeführt, so daß Bestimmungen für ein zu startendes Programm, die Angabe des Ziels für ein zu verschiebendes Programm usw. wirkungsvoll ausgeführt werden können, ohne daß Informationsaustausch mit anderen Computern erfolgt.

Unter Bezugnahme auf Fig. 16 wird nachfolgend ein viertes Ausführungsbeispiel der Erfindung beschrieben. Dieses Ausführungsbeispiel unterscheidet sich vom dritten Ausführungsbeispiel dahingehend, daß Information hinsichtlich des Funktionsvermögens und die Ausla-

stung jedes Computers im Computer selbst aufbewahrt wird. Fig. 16 zeigt die Konfiguration eines verteilten Computersystems gemäß diesem Ausführungsbeispiel. Elemente 100, 101, 110, 130—136, 140—147, 150, 151 sowie 160—166 in Fig. 16 sind identisch mit denjenigen Elementen in Fig. 1, die dort mit denselben Bezugswerten gekennzeichnet sind, wobei die entsprechenden Elemente ähnliche Vorgänge ausführen. Die Computer 102, 103 und Einzelheiten innerhalb der Computer sind in Fig. 16 weggelassen. Ferner sind die Elemente 137, 167, 290 und 291 in Fig. 16 identisch mit denjenigen Elementen, die in Fig. 11 mit denselben Bezugswerten gekennzeichnet sind, wobei die entsprechenden Elemente ähnliche Vorgänge ausführen.

Computerinformationsbereiche 330, 331 beinhalten Computerauslastung-Informationstabellen 332 bzw. 333 sowie Computerfunktionsvermögen-Informationstabellen 334 bzw. 335. Programmverschiebe-Bestimmungseinrichtungen 296, 297 bestimmen über wechselseitigen Informationsaustausch zwischen den Computern, ob die Auslastungen der jeweiligen Computer abhängig von ihren Funktionsvermögen gleich verteilt sind. Wenn die Auslastung eines Computers, der eine Programmverschiebe-Bestimmungseinrichtung enthält, größer ist als die Auslastung anderer Computer, nimmt die Programmverschiebe-Bestimmungseinrichtung eine Bestimmung zum Verschieben eines Programms in den zugehörigen Computer vor, um zu gleichmäßiger Auslastungsverteilung zu führen. Genauer gesagt, führen die Programmverschiebe-Bestimmungseinrichtungen 296, 297 diesen Vorgang unter Bezugnahme auf die Computerauslastung-Informationstabellen 332, 333 in den Computerinformationsbereichen 330 bzw. 331 aus. Wenn ein auszuführendes Programm neu angefordert wird, bestimmen die Computerauswahleinrichtungen 298, 299 über wechselseitigen Informationsaustausch zwischen den Computern, welcher Computer der Ausführung des Programms zugeordnet wird. Auch dann, wenn ein Programm von einem Computer zu einem anderen zu verschieben ist, bestimmen die Computerauswahleinrichtungen 298, 299, ebenfalls durch wechselseitigen Informationsaustausch zwischen den Computern, an welchen Computer das Programm zu verschieben ist. Die Computerauswahleinrichtungen 298, 299 nehmen diese Bestimmungen unter Bezugnahme auf die Computerauslastung-Informationstabellen 332, 333 und die Computerfunktionsvermögen-Informationstabellen 334, 335 in den Computerinformationsbereichen 330 bzw. 331 vor.

Bei diesem Ausführungsbeispiel stimmen sowohl der Vorgang zum neuen Starten eines Programms als auch der Vorgang zum Verschieben eines Programms im wesentlichen mit denjenigen des dritten Ausführungsbeispiels überein, wie sie in Verbindung mit den Fig. 11—15 beschrieben wurden. Jedoch unterscheidet sich dieses Ausführungsbeispiel vom dritten Ausführungsbeispiel im folgenden Punkt. Beim dritten Ausführungsbeispiel werden die Auslastung/Funktionsvermögen-Werte für alle Computer im System durch die Programmverschiebe-Bestimmungseinrichtung 292 oder die Computerauswahleinrichtung 294 im einzelnen Computer 100 berechnet. Bei diesem Ausführungsbeispiel werden dagegen die Auslastung/Funktionsvermögen-Werte der jeweiligen Computer 100, 101 durch die Programmverschiebe-Bestimmungseinrichtungen 296, 297 oder die Computerauswahleinrichtungen 298, 299 in den jeweiligen Computern 100, 101 berechnet, und eine Computerauswahleinrichtung in einem Computer, der



ein Programm startet, oder eine Programmverschiebe-Bestimmungseinrichtung in einem Computer, der ein Programm verschiebt, sammelt die berechneten Auslastung/Funktionsvermögen-Werte, um das Programm zu bestimmen, das zu starten ist, oder um ein Ziel zu bestimmen, an das ein Programm zu verschieben ist, und zwar unter Bezugnahme auf die so gesammelten Auslastung/Funktionsvermögen-Werte.

Zusätzlich zu den Vorteilen der gleichmäßigen Auslastungsverteilung und der gewährleisteten Echtzeiteigenschaften, wie sie beim dritten Ausführungsbeispiel vorliegen, hat dieses Ausführungsbeispiel die folgenden Vorteile. Da die Auslastungsverwaltung gleichmäßig auf die Computer verteilt ist, so daß dann, wenn ein Computer ein Programm startet oder verschiebt, die Computerauslastungsinformation nur in diesem Computer und in einem Computer verändert werden muß, in dem das gestartete Programm liegt, oder in einem Computer, an den das Programm verschoben wird, werden zunächst wirkungsvoll Zustandsänderungen der Computerauslastungsinformation wiedergegeben. Außerdem ist die Erweiterbarkeit des Systems verbessert, da die Computerinformationstabelle selbst dann nicht modifiziert werden muß, wenn ein Computer im verteilten Computersystem weggenommen wird oder ein neuer Computer hinzugefügt wird.

Es existieren Verschiedene Alternativen hinsichtlich des dritten und vierten Ausführungsbeispiels. Erstens kann, während bei diesen Ausführungsbeispielen die Auslastungsinformation, die Anzahl ausführbarer Programme anzeigt, zusammen mit der Dringlichkeit abgespeichert und ausgewertet wird, die als Kriterium zum Starten oder Verschieben eines Programms verwendet wird, der örtliche Prioritätswert in jedem Computer abgespeichert werden und anstelle der Dringlichkeit als Kriterium ausgewertet werden. Durch diese Alternative muß jeder Computer bei der Verarbeitung nur den Prioritätswert berücksichtigen, so daß die Umsetzung zwischen dem Prioritätswert und dem Dringlichkeitswert weggelassen werden kann, wodurch eine wirkungsvolle Verwaltung erzielt wird. Auch in diesem Fall können die Prioritätswert-Umsetzungseinrichtung 260, die die Entsprechungsbeziehung zwischen Dringlichkeiten und Prioritätswerten verwaltet, oder die Prioritätswert-Umsetzungseinrichtungen 144, 145 mit einer Funktion zum Modifizieren der Entsprechungsbeziehung abhängig von den Auslastungen der jeweiligen Computer versehen sein, wie für das erste Ausführungsbeispiel in Verbindung mit den Fig. 6—9 dargelegt, wodurch der Dringlichkeits- oder Prioritätswert abhängig von den Auslastungen der jeweiligen Computer eingestellt werden kann.

Während beim dritten und vierten Ausführungsbeispiel der Auslastungswert für einen Computer durch die Anzahl ausführbarer Programme wiedergegeben ist, kann der Auslastungswert auch durch die Summe von Verarbeitungsumfängen oder Vorausgesagten Verarbeitungszeiten für ausführbare Programme wiedergegeben sein. Der letztere Weg ist dahingehend von Vorteil, daß die Auslastungen jeweiliger Computer genauer abgeschätzt sind, so daß eine vollständig gleiche Auslastungsverteilung erzielt werden kann.

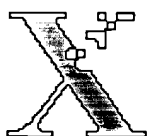
Während im verteilten Computersystem gemäß dem dritten und vierten Ausführungsbeispiel ein Computer bestimmt wird, der ein Programm startet, oder auf den ein Programm verschoben wird, wobei sowohl die Auslastung als auch das Funktionsvermögen der jeweiligen Computer berücksichtigt wird, kann die Bestimmung

nur unter Berücksichtigung einer dieser zwei Parameter erfolgen. Insbesondere in einem verteilten Computersystem, bei dem die mehreren Computer vom selben Typ sind, muß nur die Auslastung der jeweiligen Computer berücksichtigt werden. Auf diese Weise kann die Bestimmung einfacher erfolgen, wodurch der Wirkungsgrad verbessert ist.

Ferner kann, während beim verteilten Computersystem gemäß dem dritten oder vierten Ausführungsbeispiel der Auslastungswert eines Computers auf Grundlage der Anzahl ausführbarer Programme bewertet wird, wobei ein Dringlichkeitswert verwendet wird, der von null bis zum Dringlichkeitswert geht, wie er für das zu startende oder zu verschiebende Programm vergeben ist, diese Auswertung auf Grundlage der Anzahl ausführbarer Programme erfolgen, die denselben Dringlichkeitswert haben wie ein zu startendes oder zu verschiebendes Programm. Diese Auswertung ist dahingehend vorteilhaft, daß die zum Berechnen des Auslastungswerts erforderliche Zeit verringert ist, so daß ein wirkungsvollerer Betrieb erzielbar ist.

Im verteilten Computersystem können Verarbeitungsanforderungen auf solche Weise verteilt werden, daß mehrere Programme, die denselben Vorgang beinhalten, gleichzeitig auf mehreren Programmen ausgeführt werden. Herkömmlicherweise wird zum Verteilen von Verarbeitungsanforderungen ein Leerlaufprogramm (ein Programm, das keinen Vorgang ausführt) aus mehreren Programmen ausgewählt, die denselben Vorgang ausführen, und eine Verarbeitungsanforderung wird an dieses ausgewählte Programm ausgegeben. Wenn jedoch ein derartiges Programmauswahlverfahren in einem verteilten Computersystem mit Echtzeitverarbeitung realisiert wird, wird nicht immer ein Programm mit hohem Prioritätswert ausgewählt, oder es ist unbestimmt, ob der Betrieb mit einer Geschwindigkeit abgeschlossen wird, wie sie von der anfordernden Partei erwünscht ist. Fig. 17 zeigt ein fünftes Ausführungsbeispiel der Erfindung zum Auswählen eines Programms mit einem Prioritätswert, wie von einer anfordernden Partei gewünscht, aus mehreren gleichen Programmen, wobei dafür gesorgt wird, daß das ausgewählte Programm die angeforderte Verarbeitung ausführt. Dieses Ausführungsbeispiel schafft ein System zum gleichzeitigen Realisieren von Echtzeiteigenschaften zum Auswählen eines Programms, das mit einer gewünschten Geschwindigkeit arbeitet, und eine gewünschte Gleichverteilung der Auslastung. Dieses Ausführungsbeispiel ist durch die vorstehend genannte Prioritätswert-Umsetzungseinrichtung 260 und eine Programmauswahlrichtung 410 realisiert, zum Bestimmen eines Zielprogramms, an das eine Verarbeitungsanforderung auszugeben ist, wozu das Umsetzungsergebnis von der Prioritätswert-Umsetzungseinrichtung 260 verwendet wird.

In Fig. 17 sind Elemente 100—103, 110 sowie 140—143 mit denjenigen Elementen identisch, die in Fig. 1 mit denselben Bezugszahlen gekennzeichnet sind. Einzelheiten zur Hardware der Computer 100, 101 sind weggelassen. Die Prioritätswert-Umsetzungseinrichtung 260 und die Prioritätswert-Entsprechungstabelle 270 verwenden denselben Ablauf und dieselbe Datenstruktur, wie sie von ihren Gegenstücken in Fig. 10 verwendet werden. Die Benutzerprogramme 430—442 liegen in den Computern 100, 101. Unter ihnen werden die Benutzerprogramme 440—442 nicht von der Programmauswahlrichtung 410 verwaltet. Zu Programmen, die nicht der Verwaltung durch die Programmauswahlrichtung



richtung 410 unterliegen, gehören ein einzigartiges Programm im verteilten Computersystem, ein Programm, das für sich arbeitet, ohne eine Anforderungsanfrage zu empfangen usw. Jedes Benutzerprogramm erhält einen Prioritätswert und es wird in Schlangen für jeweilige Prioritätswerte in Prioritätsschlangen 140, 141 verwaltet. Blöcke 450, 451, 454—458 sowie 460—462 repräsentieren Beispiele von Programmstrukturen für die jeweiligen Anwenderprogramme, und sie verfügen über Information zu den Anwenderprogrammen 430, 431, 434—438 bzw. 440—442. Programmstrukturen, wie sie den Benutzerprogrammen 432, 433, 439 entsprechen, sind weggelassen. Bei der Erfindung werden mehrere Programme, die denselben Vorgang ausführen, gleichzeitig in mehreren Computern ausgeführt, und es wird ein Programm, für das Verarbeitung angefordert wurde, abhängig vom Dringlichkeitswert jedes Programms bestimmt, wenn eine Verarbeitungsanforderung ausgegeben wird. In Fig. 17 ist angenommen, daß die Benutzerprogramme 431—433 Programme sind, die denselben Vorgang ausführen (Programmgruppe B), während die Benutzerprogramme 435, 436 (Programmgruppe D) sowie die Benutzerprogramme 438, 439 (Programmgruppe F) entsprechend Programme sind, die denselben Vorgang ausführen.

Die Programmauswahleinrichtung 410 verwaltet die Computer, auf denen diese Programme laufen, sowie die Prioritätswerte der Programme, um ein Programm mit optimalem Prioritätswert auszuwählen. Eine Programmort-Datenbank 420 ist als Verwaltungs-Datenstruktur vorhanden, die dazu verwendet wird, ein optimales Programm auszuwählen. Die Programmort-Datenbank 420 verwaltet

Programmort-Informationsbereiche 470—479, von denen jeder dazu dient, die Kennung eines Computers abzuspeichern, wobei auf diesen Computern Programme laufen, die denselben Vorgang ausführen, und sie speichern auch den Prioritätswert des jeweiligen Programms ab. Ein alternatives Verwaltungsverfahren, bei dem Information wie eine Kommunikationsadresse für ein Programm usw. in den Programmort-Informationsbereichen abgespeichert sind, kann verwendet werden, um eine Verarbeitungsanforderung zu vereinfachen. Unter den Programmort-Informationsbereichen 470—479 sind diejenigen, die Information zu Programmen, die denselben Vorgang ausführen, zusammengefaßt, und sie werden in der Form einer Programmliste verwaltet, wie 421—426. Z.B. speichern die Programmort-Informationsbereiche 471—473 Ortsinformation zu Programmen, die denselben Vorgang ausführen (Programmgruppe B), so daß sie gemeinsam als Programmliste 422 abgespeichert sind. Übrigens können, da die Programmgruppen A, C, E nur jeweils ein Programm beinhalten, diese Gruppen ohne ihre Programmlisten in der Programmort-Datenbank 420 verwaltet werden. Wenn der Prioritätswert irgendeines in der Programmauswahleinrichtung 410 verzeichneten Programms in irgendeinem Computer geändert wird, wenn ein Programm aus einem bestimmten Computer gelöscht wird oder wenn ein zu registrierendes Programm neu gestartet wird, wird diese Programmauswahleinrichtung 410 über die Kennung des Computers, der diesem Programm zugeordnet ist, und den Prioritätswert des Programms informiert, um den Inhalt der Programmort-Datenbank 420 zu modifizieren.

Nachfolgend erfolgt eine Erläuterung für ein Verfahren zum Auswählen eines Verarbeitungsanforderungsziels auf Grundlage des Prioritätswerts unter Verwendung der Prioritätswert-Umsetzeinrichtung 260 und der

Programmauswahleinrichtung 410. Fig. 18 zeigt einen beispielhaften Ablauf für ein Programmauswahlverfahren. Bei diesem Ausführungsbeispiel ist angenommen, daß ein Benutzerprogramm 440 ein Programm in der Programmgruppe D (Benutzerprogramm 435 oder 436) anfordert, das den Dringlichkeitswert 0 spezifiziert. Zunächst fragt das Benutzerprogramm 440 aus der Programmauswahleinrichtung 410 den Ort des Programms in der Programmgruppe D ab, das dem Dringlichkeitswert 0 entspricht (Verbindung 480). Da die Programmauswahleinrichtung 410 einen Computer, in dem jedes Programm läuft, und den Prioritätswert des Programms erkennt, jedoch nicht den Dringlichkeitswert des Programms, wird der spezifizierte Dringlichkeitswert 0 in unveränderter Weise an die Prioritätswert-Umsetzeinrichtung 260 weitergegeben (Verbindung 481). Im Ergebnis wird eine Prioritätswertliste 490 entsprechend dem Dringlichkeitswert 0 an die Programmauswahleinrichtung 410 gegeben (Verbindung 482). Der Inhalt der Prioritätswertliste zeigt den Bereich örtlicher Prioritätswerte in jedem Computer, wie sie dem spezifizierten Dringlichkeitswert entsprechen. Bei diesem Ausführungsbeispiel wird die Programmauswahleinrichtung 410 mit einer Prioritätswertliste versorgt, die anzeigt, daß den Dringlichkeitswert 0 im Computer 100 die Prioritätswerte 0—1 entsprechen, und daß im Computer 101 der Prioritätswert 0 dazu gehört, was über die Verbindung 482 erfolgt. Die Programmauswahleinrichtung 410 vergleicht den Inhalt der Programmort-Datenbank 420 mit dem Inhalt der Prioritätswertliste 490, um ein Programm zu bestimmen, an das eine Verarbeitungsanforderung ausgegeben wird. Bei dem in Fig. 18 dargestellten Beispiel handelt es sich nur um Programmortinformation 476 (entsprechend dem Benutzerprogramm 436), die dem Prioritätswert entspricht, wie er durch die Prioritätswertliste 490 in der Programmliste 424 für die Programmgruppe D spezifiziert ist. Die Programmauswahleinrichtung 410 bestimmt daher, daß das im Computer 101 liegende Benutzerprogramm 436 das Ziel ist, an das die Verarbeitungsanforderung ausgegeben wird, und sie informiert das Benutzerprogramm 440 über diesem Bestimmungsergebnis (Verbindung 483). Das Benutzerprogramm 440 gibt eine Verarbeitungsanforderung an das Benutzerprogramm 436 aus, das über die Verbindung 483 darüber informiert wurde (Verbindung 484). Alternativ ist es möglich, ein Kommunikations-Zwischenschaltungsverfahren zu verwenden, bei dem die Programmauswahleinrichtung 410 Information 480 vom Benutzerprogramm 440 direkt an das Benutzerprogramm 436 überträgt (Verbindung 485).

Fig. 19 zeigt ein Beispiel für einen Verarbeitungsablauf 500 durch die Programmauswahleinrichtung 410 zum Vergleichen der Programmlisten 421—426 mit der Prioritätswertliste 490 zum Auffinden eines Programms mit einem Prioritätswert, der dem spezifizierten Prioritätswertebereich entspricht. Zunächst greift die Programmauswahleinrichtung 410 auf den ersten Programmort-Informationsbereich zu, wie er durch die spezifizierte Programmliste angegeben wird (Schritt 501). Wenn der Programmort-Informationsbereich leer ist (falls dies der Fall ist, wird dies in einem Schritt 502 ermittelt), erkennt die Programmauswahleinrichtung 410, das kein Programm vorhanden ist, das einen Prioritätswert im spezifizierten Prioritätswertebereich aufweist, und sie informiert das die die Verarbeitung anfordernde Programm über diese Tatsache (Schritt 503). Wenn der Programmort-Informationsbereich, auf den zugegriffen wurde, nicht leer ist, wird aus der Prioritäts-



wertliste 490 ein Eintrag abgerufen, der mit dem Prioritätswert eines Computers übereinstimmt, wie im Programmort-Informationsbereich abgespeichert (Schritt 504). Wenn der im Programmort-Informationsbereich abgespeicherte Prioritätswert im Prioritätswertebe-  
 5 reich enthalten ist, wie im Schritt 504 abgerufen (falls dies der Fall ist, wird es in einem Schritt 505 ermittelt), informiert die Programmauswahleinrichtung 410 das die Verarbeitung anfordernde Programm über ein Pro-  
 10 gramm, das einen Prioritätswert innerhalb des Prioritätswertebereichs aufweist (Schritt 506). Wenn im Schritt 505 bestimmt wird, daß der Prioritätswert des Programms außerhalb des spezifizierten Bereichs liegt, greift die Programmauswahleinrichtung 410 auf den  
 15 nächsten Programmort-Informationsbereich in der Programmliste zu (Schritt 507) und wiederholt dieselben Abläufe ab dem Schritt 502. Unter Verwendung des Verarbeitungsablaufs 500 wird, bei dem in Fig. 18 dar-  
 20 gestellten Beispiel, die erste Programmortinformation in einem Bereich 475 der Programmliste 424 für die Programmgruppe D zunächst mit der Prioritätswertliste 490 verglichen, und dann wird die zweite Program-  
 25 mortinformation in einem Bereich 476 mit der Prioritätswertliste 490 verglichen. Da der im Programmort-Informationsbereich 476 abgespeicherte Prioritätswert in dem durch die Prioritätswertliste 490 spezifizierten Bereich liegt, wird dieser ausgewählt und über das die  
 30 Verarbeitung anfordernde Programm informiert.

Beim fünften Ausführungsbeispiel der Erfindung kön-  
 35 nen mehrere Programme mit verschiedenen Prioritätswerten, die denselben Vorgang ausführen, verteilt in mehreren Computern im verteilten Computersystem angeordnet werden, wodurch eine Auslastungsverteilung erzielt wird, d. h. eine Verteilung von Verar-  
 40 bungsanforderungen abhängig von der Dringlichkeit. Insbesondere ist eine Wirkung dieses Ausführungsbeispiels zu erwarten, da sich sogar Verarbeitungsanfor-  
 45 derungen für denselben Inhalt der Verarbeitung häufig voneinander hinsichtlich der Dringlichkeit unterscheiden, und zwar abhängig vom anfordernden Programm.

Ferner kann gemäß dem fünften Ausführungsbeispiel durch ein Vorabestellen verschiedener Prioritätswerte für mehrere Programme, die denselben Vorgang aus-  
 50 führen, derselbe Verarbeitungsinhalt mit verschiedenen Dringlichkeitswerten angefordert werden, ohne daß die Prioritätswert-Änderungseinrichtung verwendet wird, wodurch Echtzeiteigenschaften im selben Ausmaß wie  
 55 beim ersten Ausführungsbeispiel sichergestellt sind.

Fig. 20 zeigt ein sechstes Ausführungsbeispiel der Er-  
 60 findung, das ein verteiltes Computersystem zeigt, das eine Kombination aus Prioritätswert-Umsetzeinrich-  
 65 tungen 144, 145, die verteilt in jedem Computer liegen, und eine Programmauswahleinrichtung 510 zeigt, die Kennungen der Computer speichert, auf denen Pro-  
 70 gramme, die denselben Vorgang ausführen, laufen, und die auch die Dringlichkeitswerte der Programme spei-  
 75 chert, um ein Programmanforderungsziel zu bestimmen. In den Fig. 20 sind die Elemente 100—103, 110, 140—145, 150 und 155 identisch mit denjenigen Elemen-  
 80 ten, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind. Einzelheiten zur Hardware der Computer sind weggelassen. Auch haben die Benutzerprogramme  
 85 430—442 und die Programmstrukturen 450—462 dieselben Inhalte wie in Fig. 17. Programmstrukturen, die den Benutzerprogrammen 432, 433, 439 entsprechen, sind  
 90 weggelassen.

Eine Programmauswahleinrichtung 510 verwaltet  
 95 Computer, auf denen die oben genannten Programme

laufen, und sie verwaltet die Dringlichkeitswerte der  
 Programme und bestimmt ein Programm mit optima-  
 5 lem Dringlichkeitswert. Eine Programmort-Datenbank  
 520 ist als Datenstruktur zur Verwaltung vorhanden, die  
 dazu verwendet wird, ein optimales Programm auszu-  
 wählen. Die Programmort-Datenbank 520 verwaltet  
 10 Programmort-Informationsbereiche 530—539, jeweils  
 zum Einspeichern der Kennung von Computern, auf  
 denen jeweils Programme laufen, die denselben Vor-  
 15 gang ausführen, und sie speichern den Dringlichkeits-  
 wert für das Programm. Aus den Programmort-Info-  
 mationsbereichen 530—539 werden diejenigen in der  
 Form von Programmlisten 421—426 zusammengefaßt  
 20 und verwaltet, die Information zu Programmen spei-  
 chern, die denselben Vorgang ausführen. Während die  
 Programmort-Informationsbereiche 470—479 in Fig. 17  
 jeweils die Kennung eines Computers, auf dem ein Pro-  
 25 gramm läuft und den Prioritätswert des Programms  
 speichern, speichern die Programmort-Informationsbe-  
 reiche 530—539 in Fig. 20 jeweils die Kennung eines  
 Computers, auf dem ein Programm läuft, sowie den  
 30 Dringlichkeitswert des Programms, anstelle des Priori-  
 tätsrangs. Ein alternatives Verwaltungsverfahren, das  
 Information wie Kommunikationsadressen zu einem  
 Programm usw. in den Programmort-Informationsbe-  
 35 reichen speichert, kann verwendet werden, um eine Ver-  
 arbeitungsanforderung zu vereinfachen. Wenn der  
 Dringlichkeitswert irgendeines in der Programmaus-  
 wahleinrichtung 510 registrierten Programms in irgend-  
 40 einem Computer verändert wird, wenn ein Programm  
 aus einem zugehörigen Computer gelöscht wird oder  
 ein zu registrierendes Programm neu gestartet wird,  
 wird die Programmauswahleinrichtung 510 über die  
 45 Kennung eines Computers, auf dem das betreffende  
 Programm läuft, und den Dringlichkeitswert des Pro-  
 gramms informiert, um den Inhalt der Programmlisten  
 zu modifizieren. Auch dann, wenn sich die Entspre-  
 50 chungsbeziehung zwischen dem Prioritätswert und dem  
 Dringlichkeitswert an irgendeinem Computer ändert,  
 informieren die Prioritätswert-Umsetzeinrichtungen  
 144, 145 die Programmauswahleinrichtung 510 über Än-  
 55 derungen der entsprechenden Beziehungen, um derarti-  
 ge Änderungen in den Dringlichkeitswerten widerzu-  
 spiegeln, wie sie in den Programmort-Informationsbe-  
 60 reichen 520—529 abgespeichert sind.

Fig. 20 veranschaulicht auch einen Verarbeitungsab-  
 65 lauf zum Auswählen eines Verarbeitungsanforderungs-  
 ziels auf Grundlage der Dringlichkeit, unter Verwen-  
 dung der Prioritätswert-Umsetzeinrichtungen 144, 145  
 und der Programmauswahleinrichtung 510. Bei diesem  
 70 Ausführungsbeispiel ist angenommen, daß ein Benut-  
 zerprogramm 440 eine Verarbeitungsanforderung an  
 ein Programm in der Programmgruppe D (Benutzer-  
 programm 435 oder 436) ausgibt, wie im Fall von  
 75 Fig. 18. Zunächst fragt das Benutzerprogramm 440 die  
 Programmauswahleinrichtung 510 hinsichtlich des Orts  
 eines Programms in der Programmgruppe D ab, das  
 dem Dringlichkeitswert 0 entspricht (Verbindung 540).  
 80 Da die Programmauswahleinrichtung 510 einen Com-  
 puter, auf dem jedes Programm läuft, und den Dringlich-  
 keitswerts des Programms erkennt, kann sie dann, wenn  
 sie befragt wird, den Ort eines Programms herausfinden,  
 das den Bedingungen genügt, daß es zur Programm-  
 85 gruppe D genügt und den Dringlichkeitswert 0 hat. Bei  
 diesem Ausführungsbeispiel genügt nur die Program-  
 mortinformation 536 (entsprechend dem Benutzerpro-  
 90 gramm 436) den obigen Bedingungen unter den zwei  
 Programmen, die zur Programmgruppe D gehören. Die



Programmauswahleinrichtung 410 bestimmt daher, daß das im Computer 101 liegende Benutzerprogramm 436 das Ziel ist, an das die Verarbeitungsanforderung geliefert wird, und sie informiert das Benutzerprogramm 440 über dieses Bestimmungsergebnis (Verbindung 541). Das Benutzerprogramm 440 gibt eine Verarbeitungsanforderung an das Benutzerprogramm 436 aus, das darüber über die Verbindung 541 informiert wird (Verbindung 542). Alternativ ist es auch möglich, ein Kommunikations-Zwischenschaltungsverfahren zu verwenden, bei dem die Programmauswahleinrichtung 510 Information 540 vom Benutzerprogramm 440 direkt an das Benutzerprogramm 436 überträgt (Verbindung 543).

Im verteilten Computersystem gemäß dem sechsten Ausführungsbeispiel der Erfindung kann, zusätzlich zu den Wirkungen des fünften Ausführungsbeispiels, die Wirkung erzielt werden, daß die Programmauswahl schneller als beim fünften Ausführungsbeispiel abgeschlossen wird, da die Programmauswahleinrichtung 510 die Prioritätswert-Umsetzeinrichtungen 144, 145 nicht dazu anweisen muß, einen Dringlichkeitswert in einen entsprechenden Prioritätswert umzusetzen. Da jedoch die Programmort-Informationsbereiche 530, 539 jeweils den Dringlichkeitswert eines zugeordneten Programms speichern, kann es eine Änderungen der Entsprechungsbeziehung zwischen einem Dringlichkeitswert und einem Prioritätswert wegen einer veränderten Auslastung eines bestimmten Computers, erforderlich machen, die Dringlichkeitswerte in den entsprechenden Programmort-Informationsbereichen entsprechend zu ändern. Aus diesem Grund ist das System des fünften Ausführungsbeispiels von Vorteil, wenn die Auslastungen der Computer zeitlich stark schwanken.

Abweichend von den beim fünften und sechsten Ausführungsbeispiel veranschaulichten Systemen ist es auch möglich, ein verteiltes Computersystem aufzubauen, das eine Kombination aus der Programmauswahleinrichtung 510 und der Prioritätswert-Umsetzeinrichtung 260 oder eine Kombination aus der Programmauswahleinrichtung 410 und den Prioritätswert-Umsetzeinrichtungen 144, 145 enthält. Es ist jedoch zu beachten, daß, da die in den Fig. 17, 18, 20 dargestellten Systeme keinen Fall annehmen, bei dem Programme unter Änderung ihrer Prioritätswerte ausgeführt werden, die Prioritätswert-Änderungseinrichtungen 146, 147 nicht beschrieben sind. Jedoch können diese Systeme so modifiziert werden, daß sie diese Ablaufeinrichtungen zusätzlich beinhalten, damit die Prioritätswerte von Programmen verändert werden können.

In einem verteilten Computersystem mit mehreren Programmen, die denselben Vorgang ausführen, kann, wenn eine Verarbeitungsanforderung an ein Programm ausgegeben wird, das gerade eine Verarbeitung ausführt, eine neu angeforderte Verarbeitung an eines der mehreren Programme verlegt werden, die denselben Vorgang ausführen, wodurch die jeweiligen Programme wirkungsvoll genutzt werden können, und die neu angeforderte Verarbeitung wird prompt ausgeführt, was zu einer Verbesserung der Echtzeiteigenschaften führt. Ein siebtes Ausführungsbeispiel der Erfindung zum Ausführen einer solchen Verlegung einer Verarbeitungsanforderung wird nachfolgend unter Bezugnahme auf Fig. 21 beschrieben. Fig. 21 zeigt ein verteiltes Rechnersystem mit der Funktion des Verlegens einer Verarbeitungsanforderung, die neu an ein Programm ausgegeben wird, die bereits ein anderes Programm ausführt, das denselben Vorgang wie das ursprünglich angeforderte Programm ausführt. In Fig. 21 stimmen die Elemente

100—103, 110 sowie 140—143 mit denjenigen Elementen überein, die in Fig. mit denselben Bezugszahlen gekennzeichnet sind. Details zur Hardware der Computer 100, 101 sind weggelassen. Eine Prioritätswert-Umsetzeinrichtung 260 und eine Prioritätswert-Entsprechungstabelle 270 verwenden dieselben Abläufe und Datenstrukturen wie sie bei den Gegenständen in Fig. 10 verwendet sind. Die Elemente 410, 420—426, 430—442 sowie 450—462 zeigen denselben Inhalt wie in Fig. 17. Auch sind Programmstrukturen weggelassen, die den Benutzerprogrammen 432, 433 und 439 entsprechen.

Bei diesem Ausführungsbeispiel ist jedes Programm intern mit einer Verarbeitungsverlegeeinrichtung 550 und einer Schlange 551 versehen, damit eine Verarbeitungsanforderung auf ein anderes Programm verlegt werden kann. Die Schlange 551 ist ein Puffer zum Einspeichern von Verarbeitungsanforderungen der Reihe nach, wie sie an ein zugehöriges Programm ausgegeben werden. Die Verarbeitungsverlegeeinrichtung 550 überwacht die Verarbeitungsanforderung, wie sie an der Schlange 551 antreffen, während das zugehörige Programm eine angeforderte Verarbeitung ausführt. Diese Verarbeitungsverlegeeinrichtung 550 ermöglicht es, daß Verarbeitungsanforderungen auf Programme verteilt werden können, damit die Belastung aller Computer im System gleich verteilt ist. Ein Informationsfluß 560—565 veranschaulicht, wie eine Verarbeitungsanforderung durch die Verarbeitungsverlegeeinrichtung auf ein anderes Programm verlegt wird. Dabei ist als Beispiel ein Prozeß veranschaulicht, bei dem ein Benutzerprogramm 440 eine Verarbeitungsanforderung mit dem Dringlichkeitswert 1 an ein Benutzerprogramm 438 (ein Programm in einer Programmgruppe F) ausgibt (Verbindung 560) und diese Verarbeitungsanforderung auf ein anderes Programm verlegt wird. Wenn Verarbeitungsanforderungen das Benutzerprogramm 438 erreichen, das eine Verarbeitung ausführt, werden diese Verarbeitungsanforderungen in die Schlange 551 eingespeichert. Blöcke 552, 553 repräsentieren die eintreffenden Verarbeitungsanforderungen. Wenn die Verarbeitungsverlegeeinrichtung 550 eine neue Verarbeitungsanforderung 553 empfängt, befragt sie die Programmauswahleinrichtung 410, ob ein anderes Programm mit dem Dringlichkeitswert 1, das die Verarbeitungsanforderung 553 akzeptieren soll, in der Programmgruppe F vorhanden ist (Verbindung 561). Die Programmauswahleinrichtung 410 informiert die Prioritätswert-Umsetzeinrichtung 260 über den Dringlichkeitswert 1 (Verbindung 562) und empfängt eine Prioritätswertliste 491, die den Bereich von Prioritätswerten angibt, die dem Dringlichkeitswert 1 entsprechen (Verbindung 563). In Fig. 21 sind den Computern 101, 102 entsprechende Daten aus der Prioritätswertliste 491 weggelassen. Durch Vergleichen der Prioritätswertliste 491 mit einer Programmliste 426 für die Programmgruppe F erfolgt eine Auswahl hinsichtlich Programmortinformation für ein Programm, das dem angeforderten Dringlichkeitswert genügt. Dabei vermeidet die Programmauswahleinrichtung 410 die Auswahl von Programmortinformation, die das Benutzerprogramm 438 selbst repräsentiert. Bei dem in Fig. 21 dargestellten Beispiel, wird, da die Programmortinformation 479 (entsprechend dem Benutzerprogramm 439) im Bereich der Prioritätswerte enthalten ist, wie sie durch die Prioritätswertliste 491 angegeben werden, der Ort des Benutzerprogramms 439, der der Ortsinformation entspricht, der Verarbeitungsverlegeeinrichtung 550 mitgeteilt (Verbindung 564). Die Verarbeitungsverlegeeinrichtung 550 fordert beim Emp-



fang dieser Ortsinformation das Benutzerprogramm 439 dazu auf, die neu eintreffende Verarbeitung 553 auszuführen (Verbindung 565).

Beim siebten Ausführungsbeispiel der Erfindung sind die Verarbeitungsverlegeeinrichtung 550 und die Schlange 551 in jedem Programm vorhanden. Alternativ können sie getrennt vom Programm vorliegen, um als Ablauf und als Datenstruktur zu dienen, die gemeinsam von mehreren Programmen verwendet werden können. Auch dieses Ausführungsbeispiel besteht aus dem verteilten Computersystem von Fig. 17, das eine Kombination aus der Programmauswahleinrichtung 410 und der Prioritätswert-Umsetzeinrichtung 260 sowie der zusätzlichen Verarbeitungsverlegeeinrichtung 550 beinhaltet. Zusätzlich zu dieser Konfiguration kann auch noch die Verarbeitungsverlegeeinrichtung 550 im verteilten Computersystem von Fig. 20 vorhanden sein, das eine Kombination aus der Programmauswahleinrichtung 510 und der Prioritätswert-Umsetzeinrichtungen 144, 145 enthält, um für eine ähnliche Funktion zu sorgen.

Das verteilte Computersystem des siebten Ausführungsbeispiels ermöglicht es, Verarbeitungsanforderungen, wie sie an einzelne Programme ausgegeben werden, auf andere Programme zu verlegen, die denselben Dringlichkeitswert wie die ursprünglich angeforderten Programme haben, so daß Verarbeitungsanforderungen auf mehrere Programme verteilt werden können, ohne daß eine große Anzahl von Verarbeitungsanforderungen auf ein spezielles Programm konzentriert wird. Ferner kann eine neu angeforderte Verarbeitung ausgeführt werden, ohne daß auf den Abschluß einer aktuell ausgeführten Verarbeitung gewartet werden muß, wodurch die Echtzeiteigenschaften weiter verbessert sind.

Für das siebte Ausführungsbeispiel sind alternative Konfigurationen möglich. Während das System des siebten Ausführungsbeispiels eine Verarbeitung, die von einem Programm neu angefordert wird, während dieses eine andere Verarbeitung ausführt, diese Verarbeitung auf ein anderes Programm verlegt, kann ein Programm, das die Ausführung einer neuen Verarbeitung übernehmen soll, mit einem höheren Dringlichkeitswert versehen werden, anstatt daß die neu angeforderte Verarbeitung verlegt wird, damit das Programm dazu in der Lage ist, die in Ausführung befindliche Verarbeitung und die neu angeforderte Verarbeitung schnell abzuschließen. Anders gesagt, gibt dieses System einem Programm, das mehrere Verarbeitungsanforderungen erhalten hat, abhängig von der Anzahl der Anforderungen einen höheren Dringlichkeitswert, so daß die erwünschten Echtzeiteigenschaften in vollem Umfang für die mehrere Verarbeitungsanforderungen sichergestellt sind. Es ist zu beachten, daß, damit dieses System verwendet werden kann, die Prioritätswert-Änderungseinrichtungen 146, 147 in jedem Computer vorhanden sein müssen, damit der Prioritätswert jedes Programms geändert werden kann.

#### (B) System, das eine Verarbeitungszeitgrenze verwendet

Die insoweit in Verbindung mit den Fig. 1—21 beschriebenen Ausführungsbeispiele beinhalten Verfahren, die die Dringlichkeit als gemeinsames Konzept in das verteilte Computersystem einführen, um örtliche Prioritätswerte auf gleichmäßige Weise in den jeweiligen Computern unter Verwendung des Dringlichkeitswerts zu verwalten. Wenn ein Dringlichkeitswert in einen absoluten Zahlenwert umgesetzt wird, kann für die

gleichmäßige Verwaltung eine Verarbeitungszeitgrenze verwendet werden. Die Verarbeitungszeitgrenze kann dann, wenn sie verwendet wird nicht direkt in einen Prioritätswert umgesetzt werden. Es kann an verschiedene Situationen gedacht werden. Z.B. muß, obwohl die Verarbeitungszeitgrenze lang ist, eine große Menge an Verarbeitungen ausgeführt werden; obwohl die Verarbeitungszeitgrenze kurz ist, muß nur eine kleine Menge an Verarbeitungen ausgeführt werden usw. Daher muß der Prioritätswert abhängig von der Zeitgrenze und der Verarbeitungsmenge bestimmt werden. Fig. 22 zeigt ein achttes Ausführungsbeispiel eines verteilten Computersystems zum gleichmäßigen Verwalten von Prioritätswerten unter Verwendung eines Satzes aus einer Zeitgrenze und einer Verarbeitungsmenge.

In Fig. 22 sind Elemente 100, 101, 110, 130—136, 142, 143, 146 und 147 identisch mit denjenigen Elementen, die in Fig. 1 mit denselben Bezugszahlen gekennzeichnet sind. Details zur Hardware der Computer 100, 101 wie auch der Computer 102, 103 sind weggelassen. Blöcke 570, 571 repräsentieren Prioritätsschlangen, die Programmschlangen bilden, wie sie für jeweilige Prioritätswerte in den jeweiligen Computern 100, 101 erstellt sind, um die Ausführungsreihenfolge zu verwalten. Eine Prioritätsschlange besteht aus Programmstrukturen, wofür Beispiele als 590—596 dargestellt sind. Die Programmstrukturen 590—596 beinhalten jeweils Verarbeitungsmengendaten 600 als Teil der Datenstruktur. Jede Programmstruktur ist mit einem Datenwert zur Verarbeitungsmenge versehen, wie sie von einem zugehörigen Programm auszuführen ist, um die Umsetzung von einem Satz aus einer Zeitgrenze und einer Verarbeitungsmenge in einen Prioritätswert zu unterstützen. Zu Werten, die als Verarbeitungsmenge abgespeichert werden, können die folgenden gehören:

- (1) die Anzahl von Anweisungen, die in einem Programm auszuführen sind;
- (2) die Programmgröße; und
- (3) die Verarbeitungsmenge eines Programms, wie es das vorige Mal ausgeführt wurde (tatsächliche Verarbeitungszeit des Programms, das früher ausgeführt wurde/Computerfunktionsvermögen).

Diese Werte werden alle durch die Einheit von 1.000 Anweisungen (pro Sekunde) oder dergleichen repräsentiert. In der Praxis ist das Wünschenswerteste der drei Beispiele das Beispiel (1), nämlich die Anzahl von Anweisungen, die in einem Programm auszuführen sind, jedoch ist es ziemlich schwierig, diese Anzahl von Anweisungen zu zählen. Daher werden die statische Größe eines Programms und die Verarbeitungsmenge bei der vorigen Ausführung des Programms gemessen und als Verarbeitungsmenge-Datenwert 600 abgespeichert. Um für eine gute Annäherung an die Anzahl von Anweisungen, die in einem Programm auszuführen sind, zu sorgen, wird die Programmgröße als Anfangswert für den Verarbeitungsmenge-Datenwert 600 eingespeichert, und der Datenwert 600 wird mit einem Wert für die Verarbeitungsmenge aktualisiert, wie er bei jeder Programmausführung aufgezeichnet wird.

Prioritätswert-Umsetzeinrichtungen 572, 573 entsprechen Vorgängen zum Umsetzen eines Satzes aus einer Zeitgrenze und einer Verarbeitungsmenge in einen örtlichen Prioritätswert in den zugehörigen Computern 100 bzw. 101. Bei diesem Ausführungsbeispiel beinhalten die Prioritätswert-Umsetzeinrichtungen 572, 573 Prioritätswert-Entsprechungstabellen 580, 581, die





die Übereinstimmung zwischen einer Zeitgrenze pro Einheitsverarbeitungs- und einem Prioritätswert (PRI) angeben. Die Zeitgrenze pro Einheitsverarbeitungs- und eine Zeitgrenze kann dadurch berechnet werden, daß die für ein Programm vergebene Zeitgrenze mittels der Verarbeitungs- und eine Zeitgrenze für das Programm normiert wird. Bei diesem Ausführungsbeispiel gibt die Prioritätswert-Entsprechungstabelle 580 an, daß der Computer 100 ein Programm mit dem Prioritätswert 0 in 0,017 Millisekunden pro Einheitsverarbeitungs- und eine Zeitgrenze ausführen kann, dagegen ein Programm mit dem Prioritätswert 225 in 22,77 Millisekunden pro Einheitsverarbeitungs- und eine Zeitgrenze. Die Prioritätswert-Entsprechungstabelle 581 gibt andererseits an, daß ein Computer 101 mit geringerem Funktionsvermögen länger braucht, um die Einheitsverarbeitungs- und eine Zeitgrenze abzu- und eine Zeitgrenze, und zwar, selbst wenn diese denselben Prioritätswert hat. Wegen des geringeren Funktionsvermögens führt der Computer 101 z. B. ein Programm mit einem Prioritätswert 0 in 0,034 Millisekunden pro Einheitsverarbeitungs- und eine Zeitgrenze aus, was länger als diejenige Zeit ist, die der Computer 100 benötigt, um eine Einheitsverarbeitungs- und eine Zeitgrenze eines Programms mit dem Prioritätswert 0 abzu- und eine Zeitgrenze. Die Inhalte der Prioritätswert-Entsprechungstabellen 580, 581 werden abhängig von Änderungen der Belastung in solcher Weise modifiziert, daß dieselbe Zeitgrenze einem höheren Prioritätswert bei einem stärker ausgelasteten Computer entspricht. Die Prioritätswert-Umsetzungseinrichtungen 572, 573 verwenden die Prioritätswert-Entsprechungstabellen 580, 581, um einen Satz aus einer Zeitgrenze und einer Verarbeitungs- und eine Zeitgrenze in einen zugehörigen Prioritätswert umzusetzen, und umgekehrt.

Die Prioritätswert-Entsprechungstabellen 580, 581 können durch die in Fig. 22 realisierte Feldstruktur ohne jede Änderung realisiert sein. Die Struktur kann in der Sprache C beschrieben sein, wie mit 610 in Fig. 23 dargestellt. Wenn das Feld durch die Aussage 610 erzeugt wird, sind, obwohl eine Umsetzung von einem Prioritätswert in eine entsprechende Zeitgrenze pro Einheitsverarbeitungs- und eine Zeitgrenze leicht ausgeführt werden kann, Techniken wie eine Binärsuche erforderlich, um eine Umsetzung von einer Zeitgrenze in einen entsprechenden Prioritätswert umzunehmen, da die Zeitgrenze ein kontinuierlicher Wert ist (als Gleitpunktzahl ausgedrückt). Binärsuche ist z. B. in "Information Processing Handbook" beschrieben, herausgegeben vom Institute of Information Processing Engineers, Ohm-sha, S. 58. Bei diesem Ausführungsbeispiel ist es erforderlich, für eine Zeitgrenze "Zeitgrenze" einen Prioritätswert p zu vergeben, der der folgenden Bedingung genügt:

$$\text{to\_time}[p] \leq \text{timelimit} < \text{to\_time}[p+1]$$

Daher wird ein Algorithmus auf Grundlage einer Binärsuche so modifiziert, daß eine Suche abhängig von dem in Fig. 24 dargestellten Ablauf 620 erfolgt. Der Ablauf 620 ist in der Sprache C geschrieben, und eine Umsetzung von einer Zeitgrenze auf einen entsprechenden Prioritätswert kann dadurch bewerkstelligt werden, daß der Ablauf 620 so aufgerufen wird, daß die Zeitgrenze pro Einheitsverarbeitungs- und eine Zeitgrenze als Argument angegeben ist. Verfahren zur Umsetzung zwischen der Zeitgrenze pro Einheitsverarbeitungs- und eine Zeitgrenze und einem Prioritätswert können wie folgt zusammengefaßt werden:

- (1) Umsetzung von einem Prioritätswert in eine Zeitgrenze:

timelimit-to\_time[p]

- (2) Umsetzung von einer Zeitgrenze in einen Prioritätswert:

p-to\_priority (timelimit).

Eine Verarbeitungsanforderung kann dann zurückgewiesen werden, wenn ein Computer selbst dann die Zeitgrenze nicht erfüllt, wenn er die Verarbeitung mit höchster Priorität ausführt.

Bei den in den Fig. 22—24 dargestellten Beispielen werden die Prioritätswert-Entsprechungstabellen 580, 581 dazu verwendet, einen Satz aus einer Zeitgrenze und einer Verarbeitungs- und eine Zeitgrenze in einen entsprechenden Prioritätswert umzusetzen, und umgekehrt. Alternativ können die Tabellen durch Gleichungen ersetzt werden, wie im Fall der Verwendung des Dringlichkeitswerts (siehe Fig. 3).

Nachfolgend erfolgt eine Erläuterung dafür, wie eine Verarbeitungsanforderung von einem Computer an einen anderen unter Verwendung der Prioritätswert-Umsetzungseinrichtungen 572, 573 und der Prioritätswert-Änderungseinrichtungen 146, 147, wie in diesem Ausführungsbeispiel zum Spezifizieren einer Zeitgrenze dargestellt, ausgegeben wird. Auf ähnliche Weise wie bei den Informationsübertragungen zum Spezifizieren des Dringlichkeitswerts kann ein in Echtzeit arbeitendes verteiltes Computersystem einfacher dann aufgebaut werden, wenn Informationsübertragung mit einer spezifizierten Verarbeitungszeitgrenze erfolgt. Bei diesem Ausführungsbeispiel kann daher ein Anforderungsprogramm eine Zeitgrenze spezifizieren, wenn eine Verarbeitungsanforderung ausgegeben wird. Fig. 25 veranschaulicht ein Zeitgrenze-Spezifizierverfahren gemäß der Erfindung, das in der Sprache C geschrieben ist. Eine Aussage 630 repräsentiert ein Verfahren zum Spezifizieren einer Zeitgrenze als erstes Argument in einer Verarbeitungsanforderung und eine Aussage 631 repräsentiert ein Verfahren zum Spezifizieren einer Zeitgrenze als letztes Argument in einer Verarbeitungsanforderung. Während ein Satz aus einer Zeitgrenze und einer Verarbeitungs- und eine Zeitgrenze bestimmt werden müssen, um einen entsprechenden Prioritätswert festzulegen, wie oben beschrieben, muß die Zeitgrenze nur bei Ausgeben einer Verarbeitungsanforderung spezifiziert werden, da die Verarbeitungs- und eine Zeitgrenze dadurch bestimmt werden kann, daß auf den Verarbeitungs- und eine Zeitgrenze-Datenwert 600 in der Programmstruktur im angeforderten Computer Bezug genommen wird.

Fig. 26 zeigt schematisch einen Verarbeitungsablauf zum Ausgeben einer Verarbeitungsanforderung unter Verwendung des oben genannten Zeitgrenze-Spezifizierverfahrens. Elemente 100, 101, 110, 130—136, 142, 143, 146 und 147 in Fig. 26 sind mit solchen Elementen identisch, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind. Die Computer 102, 103 und Einheiten innerhalb derselben sind weggelassen. Auch sind die Elemente 570—573, 580, 581, 590—596, 600 in Fig. 26 identisch mit Elementen, die in Fig. 22 mit denselben Bezugswerten gekennzeichnet sind. Bei diesem Ausführungsbeispiel zeigt der Ablauf in Fig. 26, daß ein Benutzerprogramm 133 eine Verarbeitungsanforderung an ein Benutzerprogramm 136 ausgibt. Wenn es das Benutzerprogramm 133 wünscht, der vom Benutzerprogramm 136 auszuführenden Verarbeitung eine bestimmte Zeitgrenze aufzuerlegen, führt das Benutzerprogramm 136 eine die Zeitgrenze spezifizierende In-



formationsübertragung 630 oder 631 vor (Verbindung 640). In Fig. 26 ist angenommen, daß eine Zeitgrenze von 5 Millisekunden spezifiziert ist. Der Computer 101 muß, wenn er Information zum Spezifizieren der Zeitgrenze empfängt, diese Zeitgrenze in einen örtlichen Prioritätswert in ihm Umsetzen und das Benutzerprogramm abhängig vom Prioritätswert ausführen, wie er sich aus der Umsetzung ergibt. Das Benutzerprogramm 136 fordert die Prioritätswert-Änderungseinrichtung 147 dazu auf, die spezifizierte Zeitgrenze von 5 Millisekunden in einen entsprechenden Prioritätswert umzusetzen (Verbindung 641). Die Prioritätswert-Änderungseinrichtung 147 fordert die Prioritätswert-Umsetzungseinrichtung 573 dazu auf, die Zeitgrenze in einen entsprechenden Prioritätswert umzusetzen (Verbindung 642). Die Prioritätswert-Umsetzungseinrichtung 573 ruft den Verarbeitungsmenge-Datenwert 600 aus einer 5 dem Benutzerprogramm 136 zugehörigen Programmstruktur 596 ab (Schritt 643) und bestimmt aus der abgerufenen Verarbeitungsmenge und der spezifizierten Zeitgrenze einen Prioritätswert. Die Verarbeitungsmenge ist für das Benutzerprogramm 136 zu 64.000 Anweisungen angenommen. Dann werden aus der Zeitgrenze von fünf Millisekunden und der Verarbeitungsmenge von 64.000 Anweisungen 0,78 Millisekunden als Zeitgrenze pro Einheitsverarbeitungsmenge berechnet (Berechnung 644). Unter Bezugnahme auf die Prioritätswert-Entsprechungstabelle 581 ergibt sich, daß der Prioritätswert, der dem 5 Wert entspricht, wie er sich durch die Berechnung 644 ergab, eins ist. Die Prioritätswert-Umsetzungseinrichtung 573 informiert die Prioritätswert-Änderungseinrichtung 147 über den Prioritätswert 1 als Umsetzungsergebnis, um anzugeben, daß das Benutzerprogramm 136 mit dem Prioritätswert 1 ausgeführt werden muß (Verbindung 645). Schließlich verlegt die Prioritätswert-Änderungseinrichtung 147 die Programmstruktur 536 in einer Prioritätsschlange 571 auf eine Schlange für den Prioritätswert 1 (Verbindung 646).

Auf die vorstehend genannte Weise können die Prioritätswerte der Programme, die in verschiedenen Computern liegen, gleichmäßig auf Grundlage des gemeinsamen Konzepts der Zeitgrenze verwaltet werden. Übrigens ist es auch möglich, wie im Fall eines verteilten Computersystems unter Verwendung von Dringlichkeitswerten, daß das Benutzerprogramm 133 die Prioritätswert-Umsetzungseinrichtung 572 dazu anweist, die Zeitgrenze aus dem eigenen Prioritätswert 4 und der Verarbeitungsmenge zu berechnen, und es die Verarbeitungsanforderung 640 mit dieser spezifizierten Zeitgrenze ausgibt.

Ähnlich wie beim verteilten Rechnersystem, das Dringlichkeitswerte verwendet, um die örtlichen Prioritätswerte in den jeweils zugehörigen Computern zu verwalten, kann der Inhalt der Prioritätswert-Entsprechungstabelle abhängig von Änderungen der Auslastung auch dann verändert werden, wenn für eine solche Verwaltung die Zeitgrenze verwendet wird. Die folgenden zwei Vorgehensweisen werden beispielhaft als Verfahren zum Umändern einer Entsprechungstabelle zwischen einer Zeitgrenze und einem Prioritätswert erläutert:

(1) Vorgehensweise, gemäß der die Zeitgrenze in der Prioritätswert-Entsprechungstabelle proportional zur sich verändernden Auslastung verändert wird; und

(2) Vorgehensweise, gemäß der die Zeitgrenze in der Prioritätswert-Entsprechungstabelle unter

Verwendung der Differenz zwischen einer tatsächlichen Ausführungszeit und der Zeitgrenze verändert wird.

Fig. 27 zeigt einen Ablauf für die erste vorgehensweise 650 zum Modifizieren der Prioritätswert-Entsprechungstabelle. Der Auslastungswert zum Zeitpunkt, zu dem die Prioritätswert-Entsprechungstabelle das letztmal modifiziert wurde, wird als voriger Auslastungswert bezeichnet, und ein anschließender Änderungswert der Auslastung wird als Gesamtauslastungsänderung bezeichnet. Wenn sich die Auslastung ändert, wird die aktuelle Auslastungsänderung zur Gesamtauslastungsänderung addiert (Schritt 651) und es wird ermittelt, ob die sich ergebende Gesamtauslastungsänderung einem vorgegebenen Wert entspricht, oder größer ist als dieser, und zwar unabhängig von der Änderungsrichtung (d. h., es wird ermittelt, ob der Absolutwert der Gesamtauslastungsänderung dem vorgegebenen Wert entspricht oder größer ist) (Schritt 652). Wenn die Gesamtauslastungsänderung dem vorgegebenen Wert entspricht oder größer ist, wird die Zeitgrenze in der Prioritätswert-Entsprechungstabelle abhängig vom Verhältnis aus dem vorigen Belastungswert und dem aktuellen Belastungswert korrigiert (Schritt 653). Wenn der Inhalt der Prioritätswert-Entsprechungstabelle modifiziert ist, wird die Gesamtauslastungsänderung auf null gesetzt, und der vorige Auslastungswert wird auf den aktuellen Auslastungswert gesetzt, was als abschließender Schritt erfolgt (Schritt 654), woraufhin die Messung der Auslastungsänderung wieder aufgenommen wird.

Fig. 28 zeigt Tabellen zum Erläutern von Bedeutungen bei der Modifizierung der Prioritätswert-Entsprechungstabelle abhängig vom Verhältnis des vorigen Auslastungswerts zum aktuellen Auslastungswert, wie im Schritt 653 in Fig. 27 ausgeführt. Dabei ist angenommen, daß die Prioritätswert-Entsprechungstabelle 580 Daten enthält, die auf Grundlage eines Lastwerts 50 berechnet wurden (d. h., daß die Anzahl ausführbarer Programme 50 ist), und die Prioritätswert-Entsprechungstabelle wird jedesmal dann modifiziert, wenn der Auslastungswert um zehn erhöht oder verringert ist. Wenn die Auslastung um zehn erhöht ist, wird die Prioritätswert-Entsprechungstabelle so modifiziert, wie es in einer Tabelle 660 dargestellt ist. Genauer gesagt, wird jeder Zeitgrenzewert in der Tabelle 660 dadurch hergeleitet, daß der Zeitgrenzewert in der ursprünglichen Tabelle 580 mit dem Verhältnis aus dem vorigen Auslastungswert 50 zum geänderten Auslastungswert 60 multipliziert wird. Auf ähnliche Weise kann dann, wenn der Auslastungswert um zehn verringert ist, eine Prioritätswert-Entsprechungstabelle 661 dadurch erzeugt werden, daß jeder Zeitgrenzewert in der ursprünglichen Tabelle 580 mit dem Verhältnis aus dem vorigen Auslastungswert 50 zum veränderten Auslastungswert 40 multipliziert wird. Auf diese Weise kann die demselben Prioritätswert entsprechende Zeitgrenze dann verlängert werden, wenn die Auslastung größer ist (eine längere Zeit ist erforderlich, um die Verarbeitung abzuschließen), wohingegen eine Verkürzung möglich ist, wenn die Auslastung kleiner ist (die Verarbeitung wird in kürzerer Zeit abgeschlossen).

Nachfolgend wird die zweite Vorgehensweise zum Modifizieren der Prioritätswert-Entsprechungstabelle unter erneuter Bezugnahme auf Fig. 26 erläutert. Bei der zweiten Vorgehensweise empfängt die Prioritätswert-Umsetzungseinrichtung 573 nach dem Ausführen des Programms 136 mit dem auf eins geänderten Prioritätswert





tätswert die tatsächliche Ausführungszeit für das Benutzerprogramm 136 (Verbindung 647). Bei dem in Fig. 26 dargestellten Beispiel ist angenommen, daß die tatsächliche Verarbeitungszeit 3,47 Millisekunden beträgt. Dann wird die Ausführungszeit pro Einheitsverarbeitungszeit zu 0,54 Millisekunden berechnet ( $3,47/4 = 0,87$ ) (Berechnung 648). Dieser Wert wird unmittelbar als dem Prioritätswert 1 entsprechende Zeitgrenze abgespeichert, um die Prioritätswert-Entsprechungstabelle 581 zu korrigieren (Schritt 649). Es ist zu beachten, daß zum tatsächlichen Korrigieren der Prioritätswert-Entsprechungstabelle ein vorgegebener Wert (z. B. 0,1 Millisekunden) zum berechneten Wert addiert werden kann, d. h. 0,54 Millisekunden, um für Sicherheit zu sorgen.

Vorstehend wurde das verteilte Computersystem gemäß dem achten Ausführungsbeispiel zum gleichmäßigen Verwalten örtlicher Prioritätswerte, wie sie in den jeweils zugehörigen Computern festgelegt sind, beschrieben. Bei diesem Ausführungsbeispiel ist das Konzept einer Zeitgrenze verwendet, um gleichmäßige Verwaltung der örtlichen Prioritätswerte in den mehreren Computern im verteilten Computersystem zu ermöglichen, die sich voneinander hinsichtlich des Funktionsvermögens, der Auslastung und des Typs unterscheiden. Da die Zeitgrenze in jedem Computer abhängig vom Funktionsvermögen und der Auslastung desselben dadurch in einen entsprechenden Prioritätswert umgesetzt wird, daß nur eine Verarbeitungsanforderung mit spezifizierter Zeitgrenze ausgegeben wird, ist es möglich, ein in Echtzeit arbeitendes verteiltes Computersystem auf einfache Weise aufzubauen.

Ferner kann gemäß diesem Ausführungsbeispiel, da ein Programm von einem Computer ausgeführt wird, nachdem die Zeitgrenze und die Verarbeitungsmenge hierfür in einen entsprechenden Prioritätswert im Computer abhängig vom Funktionsvermögen und der Auslastung des Computers umgesetzt wurden, eine neu angeforderte Verarbeitung mit relativ kurzer Zeitgrenze vor einer zuvor zugeordneten Verarbeitung mit langer Zeitgrenze ausgeführt werden, was es ermöglicht, Echtzeitverarbeitung strenger einzuhalten, während die Vorteile mehrerer Computer wirkungsvoll genutzt werden.

Ferner kann bei diesem Ausführungsbeispiel der Inhalt der Prioritätswert-Entsprechungstabelle so verändert werden, daß der Prioritätswert in jedem Computer abhängig von dynamischen Auslastungsänderungen verwaltet werden kann.

Ferner kann bei diesem Ausführungsbeispiel die Zeitgrenze abhängig von der Bedeutung einer Verarbeitung spezifiziert werden, wenn eine Verarbeitungsanforderung von einem Computer auf einen anderen verlegt wird. Außerdem ist es zum Spezifizieren der Zeitgrenze nicht erforderlich, das Funktionsvermögen, die Auslastung und den Typ eines Computers zu kennen, an den eine Anforderung geht.

Während das System dieses Ausführungsbeispiels die Auslastung eines Computers durch die Anzahl ausführbarer Programme repräsentiert, kann die Auslastung durch die Summe von Verarbeitungsmengen oder der vorausgesagten Verarbeitungszeit ausführbarer Programme repräsentiert werden. Der letztere Weg ist dahingehend von Vorteil, daß die Auslastungen jeweiliger Computer genauer abgeschätzt sind, so daß Echtzeiteigenschaften strenger eingehalten werden können. Während das Systems dieses Ausführungsbeispiels den Prioritätswert unter Berücksichtigung sowohl des Funktionsvermögens als auch der Auslastung jedes Compu-

ters bestimmt, kann nur einer dieser Parameter bei der Bestimmung des Prioritätswerts berücksichtigt werden. Insbesondere bei einem verteilten Computersystem, das nur Computer desselben Typs enthält, muß nur die Auslastung berücksichtigt werden, wodurch der Wirkungsgrad der Umsetzung zwischen Zeitgrenzen und Prioritätswerten verbessert ist.

Um die örtlichen Prioritätswerte von Computern gleichmäßig zu verwalten, die zu einem verteilten Computersystem gehören, wobei das Konzept der Zeitgrenze verwendet wird, ermöglicht eine Berücksichtigung der Verbindungszeit zwischen Computern eine genauere Prioritätsverwaltung. Fig. 29 zeigt ein neuntes Ausführungsbeispiel der Erfindung, das die Verbindungszeit zwischen Computern bei der Prioritätsverwaltung berücksichtigt. Die Elemente 100, 101, 110, 130—136, 142, 143, 146 und 147 in Fig. 29 sind identisch mit den Elementen, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind. Die Computer 102, 103 sowie Einzelheiten innerhalb derselben sind weggelassen. Die Elemente 570—573, 580, 581, 590—596 sowie 600 in Fig. 29 sind identisch mit denjenigen Elementen, die in Fig. 22 mit denselben Bezugswerten gekennzeichnet sind. Auch sind der Ablauf der Umsetzung zwischen einer Zeitgrenze und einem Prioritätswert innerhalb der Verbindungen 641, 642 sowie 645—646 identisch mit denjenigen, die in Fig. 26 dargestellt sind.

Das System dieses Ausführungsbeispiels ist mit Verbindungszeit-Voraussageeinrichtungen 670, 671 versehen, um die Verbindungszeit zwischen Computern voraussagen. Die Verbindungszeit-Voraussageeinrichtungen 670, 671 arbeiten unabhängig von den Prioritätswert-Umsetzungseinrichtungen 572, 573, den Prioritätswert-Änderungseinrichtungen 146, 147 usw. Die Verbindungszeit-Voraussageeinrichtungen 670, 671 verfügen intern über Verbindungszeittabellen 680, 681, so daß sie dann, wenn eine Computerkennung vorgegeben wird, aus den Verbindungszeittabellen 680 bzw. 681 die Verbindungszeit zwischen dem zugehörigen Computer und demjenigen Computer ausgeben, für den die Kennung eingegeben wurde. Um die Verbindungszeittabellen 680, 681 zu errichten, kann beispielsweise eines der zwei folgenden Verfahren verwendet werden:

- (1) beim Aktivieren eines Computers nimmt dieser Verbindung mit allen restlichen Computern auf, um die Ausführungszeit für die Verbindung abzuspeichern; oder
- (2) die Verbindungstabelle wird erst dann errichtet, wenn eine Verbindungszeitvoraussage angefordert wird. Wenn dies das erste Mal erfolgt, nimmt der Computer Verbindung mit dem angeforderten Computer auf, um dann die Ausführungszeit für die Verbindung abzuspeichern.

Fig. 29 veranschaulicht auch einen Ablauf für Informationsaustausch zum Ausgeben einer Verarbeitungsanforderung unter Verwendung der Verbindungszeit-Voraussageeinrichtungen 670, 671. Ähnlich wie Fig. 26 zeigt Fig. 29 ein Beispiel, bei dem ein Benutzerprogramm 133 ein anderes Benutzerprogramm 136 dazu auf fordert, eine Verarbeitung innerhalb einer spezifizierten Zeitgrenze auszuführen. Es sei angenommen, daß das Benutzerprogramm 133 der vom Benutzerprogramm 136 auszuführenden Verarbeitung eine Zeitgrenze von 8,14 Millisekunden auferlegen möchte. Zu diesem Zweck führt das Benutzerprogramm 133 einen Informationsaustausch 690 aus, um die Zeitgrenze in der



durch 630 oder 631 in Fig. 25 dargestellten Form zu spezifizieren. Diese Zeitgrenze wird als solche angesehen, die die Zeit beinhaltet, die für Kommunikation zwischen den Computern erforderlich ist, gesehen vom Benutzerprogramm 133 aus. Genauer gesagt, wird die Zeitgrenze, wie sie vom Benutzerprogramm 136 aus zu sehen ist, dadurch hergeleitet, daß die Zeit, wie sie für die Verbindung zwischen den Computern erforderlich ist, von der Zeitgrenze von 8,14 Millisekunden abgezogen wird. Das Benutzerprogramm 136 befragt die Verbindungszeit-Voraussageeinrichtung 671 hinsichtlich der Verbindungszeit zum Computer 100 (Verbindung 691). Daraufhin nimmt die Verbindungszeit-Voraussageeinrichtung 671 auf die Verbindungszeitabelle 681 Bezug und liefert die Verbindungszeit zum spezifizierten Computer 100 an das Benutzerprogramm 136 zurück (Verbindung 692). Bei den beispielhaften Zahlenwerten in Fig. 29 beträgt die Verbindungszeit zum Computer 100 1,57 Millisekunden. Das Benutzerprogramm 136 kann die auferlegte Zeitgrenze dadurch herausfinden, daß es die Verbindungszeit von der spezifizierten Zeitgrenze abzieht. Bei diesem Beispiel werden 5 Millisekunden als Zeitgrenze für das Benutzerprogramm 136 dadurch erhalten, daß die Verbindungszeit ( $1,57 \times 2 = 3,14$  Millisekunden) (zum Aussenden einer Verarbeitungsanforderung und zum Rückliefern des Ergebnisses ist dieselbe Verbindungszeit erforderlich) von 8,14 Millisekunden abgezogen wird. Die anschließenden Schritte sind denen in Fig. 26 ähnlich.

Beim vorstehend beschriebenen Verarbeitungsablauf wird die endgültige Zeitgrenze durch das die Verarbeitung anfordernde Programm berechnet, das die Verbindungszeit von der spezifizierten Zeitgrenze abzieht. Alternativ kann das die Verarbeitung anfordernde Programm aus der Verbindungszeit-Voraussageeinrichtung 870 eine Verbindungszeit abfragen, um diese vorab von der spezifizierten Zeitgrenze abzuziehen.

Beim vorstehend beschriebenen neunten Ausführungsbeispiel kann die beim achten Ausführungsbeispiel verwendete Zeitgrenze genauer in einen Prioritätswert umgesetzt werden. Dies ist insbesondere dann von Wirkung, wenn die Verbindungszeit zwischen Computern zu lang ist, als daß sie vernachlässigt werden könnte.

Bei den Systemen gemäß dem achten und neunten Ausführungsbeispiel sind die Prioritätswert-Umsetzeinrichtungen 572, 573 gesondert in den jeweiligen Computern vorhanden. Alternativ kann eine Prioritätswert-Umsetzeinrichtung gemeinsam in einem einzelnen Computer vorhanden sein, wie in Fig. 10 dargestellt.

Nachfolgend wird ein verteiltes Computersystem gemäß einem zehnten Ausführungsbeispiel der Erfindung zum Ausführen einer Auslastungsverteilung abhängig von der Wichtigkeit jedes Programms gegeben. Beim zehnten Ausführungsbeispiel wird die Auslastungsverteilung auf Grundlage einer Zeitgrenze vorgenommen, um eine Situation zu vermeiden, bei der Programme mit kurzen Zeitgrenzen in einem einzelnen Computer konzentriert werden. Fig. 30 zeigt die Konfiguration des verteilten Computersystems gemäß dem zehnten Ausführungsbeispiel. Die Elemente 100, 101, 110, 130—136, 142, 143, 146 und 147 in Fig. 30 sind identisch mit denjenigen Elementen, die in Fig. 1 mit denselben Bezugszahlen gekennzeichnet sind. Die Computer 102, 103 sowie Einzelheiten innerhalb derselben sind weggelassen. Auch sind die Elemente 137, 290, 291 in Fig. 30 identisch mit denjenigen, die in Fig. 16 mit denselben Bezugszahlen gekennzeichnet sind, und entsprechende Elemente führen ähnliche Vorgänge aus. Prioritätsschlangen 570,

571, Prioritätswert-Umsetzeinrichtungen 572, 573, Prioritätswert-Entsprechungstabellen 580, 581, Programmstrukturen 590—596 sowie der Verarbeitungsmenge-Datenwert 600 in Fig. 30 sind identisch mit den entsprechenden Einheiten in Fig. 22. Ein Benutzerprogramm 137 wird durch eine Programmstruktur 597 verwaltet.

Computerinformationsbereiche 330, 331 beinhalten Computerauslastung-Informationstabellen 370 bzw. 371 sowie Computerfunktionsvermögen-Informationstabellen 334 bzw. 335. Die Computerauslastung-Informationstabellen 370, 371 beinhalten Auslastungswerte für jeweilige Computer für jede Zeitgrenze pro Einheitsverarbeitungs- und Funktionsmenge. Während bei diesem Ausführungsbeispiel Begrenzer für die Zeitgrenze pro Einheitsverarbeitungs- und Funktionsmenge in der Computerauslastungs- und Funktionsvermögen-Information nicht mit regelmäßigen Intervallen vorhanden sind, können sie mit regelmäßigen Intervallen vorliegen, um die Auslastungsverteilung zu vereinfachen.

Unter Bezugnahme auf die Fig. 30—34 wird nun der Betrieb des Systems gemäß diesem Ausführungsbeispiel beschrieben. Zunächst wird erläutert, wie ein neues Programm gestartet wird. Wenn der Benutzer einen Befehl eingibt, der eine ausführbare Datei eines Programms zusammen mit einer Zeitgrenze spezifiziert, analysiert das Betriebssystem den Befehl und fordert die jeweiligen Programmausführungs- und Programmverschiebeeinrichtungen 290, 291 in den Betriebssystemen dazu auf, das Programm zu starten, wodurch dieses tatsächlich gestartet werden kann. Bei diesem Beispiel ist angenommen, daß der Computer 100 einen Befehl zum Starten eines Programms mit einer auf 50 Millisekunden eingestellten Zeitgrenze empfängt, und daß die Programmausführungs- und Programmverschiebeeinrichtung 290 daraufhin denjenigen Vorgang ausführt, wie er zum Starten eines Programms gehört. Die folgende Beschreibung läuft gemäß diesem Szenarium ab.

Zunächst wird unter Bezugnahme auf Fig. 31 ein Verarbeitungsablauf 340 für einen Programmstart erläutert. Die Programmausführungs- und Verschiebeeinrichtung 290 fordert eine Computerauswahl- und Verschiebeeinrichtung 298 dazu auf, zu bestimmen, mit welchem Zeitgrenzenbereich ein neu zu startendes Programm innerhalb der Computerauslastungs- und Funktionsvermögen-Information enthalten ist (Schritt 341). Wenn z. B. angenommen wird, daß die Verarbeitungsmenge eines Programms zu 128.000 Anweisungen abgeschätzt ist, wird die Zeitgrenze pro Einheitsverarbeitungs- und Funktionsmenge zu  $0,39$  Millisekunden ( $50/128 = 0,39$ ) berechnet. In den Computerauslastung- und Funktionsvermögen-Informationstabellen 370, 371 fallen die  $0,39$  Millisekunden in den Zeitgrenzenbereich  $0,05—1,00$ . Danach werden die Computerauswahl- und Verschiebeeinrichtungen 298, 299 dazu aufgefordert, die Auslastungs- und Funktionsvermögen-Werte im Bereich der Zeitgrenze pro Einheitsverarbeitungs- und Funktionsmenge von  $0$  Millisekunden bis zu dem im Schritt 341 bestimmten Bereich zu berechnen (Schritt 342). Die Berechnung wird dadurch ausgeführt, daß ein Auslesen aus den Computerauslastung- und Funktionsvermögen-Informationstabellen 370, 371 und den Computerfunktionsvermögen-Informationstabellen 334, 335 erfolgt, wie in Fig. 30 dargestellt. Die Fig. 32 stellt Tabellen dar, die die Auslastungs- und Funktionsvermögen-Information, den Auslastungs- und Funktionsvermögen-Wert jedes Computers beinhalten, in den Zustand versetzt, wie in Fig. 30 dargestellt. Die Computerauslastung- und Funktionsvermögen-Informationstabellen 370, 371 geben jeweils die Anzahl ausführbarer Programme für die zugehörigen Computer 100, 101 für jeden Bereich der Zeitgrenze pro Einheitsverarbeitungs- und Funktionsmenge an, während die Zahlen der ausführbaren Programme in den jeweiligen Zeitgrenze-



bereichen von 0 Millisekunden bis zu speziellen Zeitgrenzwerten (0–0,05; 0–1,00; 0–3,00; 0–9,00; ...; 0–200) für die jeweiligen Computer 100, 101 in den Auslastungswerttabellen 380, 381 abgespeichert sind. Die Anzahl der ausführbaren Programme wird durch das jeweilige Computerfunktionsvermögen (Information) geteilt, um die Auslastungs/Funktionsvermögen-Werte herzuleiten, wie sie in den Tabellen 382, 383 abgespeichert sind. Da das hier betrachtete Programm neu gestartet wird, ist ihm eine Zeitgrenze von 0,39 Millisekunden auferlegt, und der Auslastungs/Funktionsvermögen-Wert, wie er dem minimalen Zeitgrenzbereich 0–1 Millisekunde entspricht, der 0,39 Millisekunden enthält, wird für den Computer 100 zu 1,10 und für den Computer 101 zu 0,67 berechnet.

Danach kommunizieren die Computerauswahleinrichtungen 298, 299 miteinander, um den Computer mit dem kleinsten Auslastungs/Funktionsvermögen-Wert als denjenigen Computer auszuwählen, dem die Ausführungen des zu startenden Programms zugeordnet wird (Schritt 343). Unter Bezugnahme auf Fig. 32, aus der Tabellen für die mit 102 und den anschließenden Bezugswerten bezeichneten Computer weggelassen sind, wird unter der Annahme des Auslastungs/Funktionsvermögen-Werts von 0,67 für den Computer 101 als kleinster Wert dieser Computer 101 ausgewählt. Schließlich führt die Programmausführungs- und Programmverschiebeeinrichtung den Vorgang zum Starten des Programms durch den ausgewählten Computer aus (Schritt 344). Dieser Vorgang wird in solcher Weise erzielt, daß die Programmausführungs- und Programmverschiebeeinrichtung des Computers, der den Programmstartablauf gestartet hat, die Programmausführungs- und Programmverschiebeeinrichtung des ausgewählten Computers dazu auffordert, das Programm in einen ausführbaren Zustand zu überführen. Bei diesem Beispiel kommuniziert die Programmausführungs- und Programmverschiebeeinrichtung 290 des Computers 100 mit der Programmausführungs- und Programmverschiebeeinrichtung 291 des Computers 101, um diese dazu aufzufordern, das Programm in einen ausführbaren Zustand zu bringen.

Anschließend wird ein Verarbeitungsablauf zum Verschieben eines ausführbaren Programms erläutert. Bei diesem Ausführungsbeispiel werden die Programmverschiebe-Bestimmungseinrichtungen 296, 297 periodisch so gestartet, daß sie miteinander kommunizieren, um die Auslastungen der ihnen jeweils zugeordneten Computer klarzustellen. Die Programmverschiebe-Bestimmungseinrichtungen 296, 297 bestimmen, wenn sie herausfinden, daß die Auslastungen der jeweiligen Computer unter Berücksichtigung ihrer Funktionsvermögen nicht gleichmäßig verteilt sind, daß ein Programm in solcher Weise verschoben wird, daß die jeweiligen Computer gleich ausgelastet sind, und sie fordern die Programmausführungs- und Programmverschiebeeinrichtungen 290, 291 dazu auf, ein Programm zu verschieben. Diese Programmausführungs- und Programmverschiebeeinrichtungen 290, 291 fordern ihrerseits die Computerauswahleinrichtungen 298, 299 dazu auf, einen Zielcomputer zu bestimmen und einen Vorgang auszuführen, wie er der Verschiebung eines Programms zugeordnet ist, und zwar in Zusammenarbeit mit der Programmausführungs- und Programmverschiebeeinrichtung des Zielcomputers. Bei diesem Ausführungsbeispiel werden die obengenannten Auslastungs/Funktionsvermögen-Werte dazu verwendet, zu ermitteln, ob die Auslastungen der jeweiligen Computer angesichts

ihrer Funktionsvermögen gleich verteilt sind.

Unter Bezugnahme auf Fig. 33 wird ein Verarbeitungsablauf 350 zum Verschieben eines Programms erläutert. Zunächst setzt die Programmverschiebe-Bestimmungseinrichtung eine einen Zeitgrenzbereich repräsentierende Variable auf 0,05 Millisekunden (Schritt 351). Dieser Wert dient als Obergrenze eines Bereichs, in dem die Zeitgrenze pro Einheitsverarbeitungs menge den kleinsten Wert innerhalb der Computerauslastungs-information 370–371 hat. Dann wird der Auslastungs/Funktionsvermögen-Wert für jeden Computer im Zeitgrenzbereich von 0 bis t Millisekunden berechnet (Schritt 352). Wie oben beschrieben, sind die für die in Fig. 30 veranschaulichten Zustände berechneten Auslastungs/Funktions-Vermögen-Werte für die Computer 100, 101 diejenigen, die in den Tabellen 382, 383 aufgelistet sind. Danach wird die Differenz D zwischen dem größten Auslastungs/Funktionsvermögen-Wert und dem kleinsten Auslastungs/Funktionsvermögen-Wert unter den Auslastungs/Funktionsvermögen-Werten, wie sie für alle Computer berechnet wurden, bestimmt (Schritt 353). Obwohl die Information zu den mit 102 und mit höheren Bezugswerten gekennzeichneten Computern aus Fig. 32 weggelassen ist, sei angenommen, daß der Auslastungs/Funktionsvermögen-Wert von 0,33 für den Computer 101 der größte Wert und der Auslastungs/Funktionsvermögen-Wert von 0,30 für den Computer 100 der kleinste Wert ist, wenn t 0,05 Millisekunden ist; die Differenz D ist dann zu 0,03 berechnet. Dann wird untersucht, ob dieser Wert der Differenz D einen Schwellenwert Dt überschreitet (Schritt 354). Bei diesem Ausführungsbeispiel ist angenommen, daß der Schwellenwert Dt auf 0,3 eingestellt ist. Wenn die Differenz D den Schwellenwert Dt überschreitet, geht der Ablauf zu einem Schritt 355 über, andernfalls zu einem Schritt 358. Da bei diesem Beispiel die Differenz D den Schwellenwert Dt nicht überschreitet, wenn n den Wert 0 hat, geht der Ablauf zum Schritt 358 weiter, wo ermittelt wird, ob der Wert von t der Maximalwert im Zeitgrenzbereich ist (der Maximalwert beträgt beim in Fig. 30 veranschaulichten Beispiel 200 Millisekunden). Wenn der Wert t der Maximalwert im Zeitgrenzbereich ist, endet der Ablauf. Falls nicht, geht der Ablauf zu einem Schritt 359 weiter, in dem die Obergrenze des nächsten Zeitgrenzbereichs in die Variable t eingesetzt wird. Bei diesem Beispiel geht, da t nun auf 0,05 Millisekunden eingestellt ist, der Ablauf zum Schritt 359 über, in dem t auf eine Millisekunden gesetzt wird.

Dann springt der Ablauf zum Schritt 352 zurück, um die Auslastungs/Funktionsvermögen-Werte für die jeweiligen Computer zu berechnen, und dann wird zum Schritt 353 weitergegangen, in dem die Differenz D berechnet wird. Obwohl die Information für die mit 102 und höheren Bezugswerten gekennzeichneten Computer in Fig. 32 weggelassen ist, sei angenommen, daß der Auslastungs/Funktionsvermögen-Wert von 1,10 für den Computer 100 der größte und der Auslastungs/Funktionsvermögen-Wert von 0,67 für den Computer 101 der kleinste Wert ist, wenn t eine Millisekunde ist; dann wird die Differenz D zu 0,43 berechnet. Wenn t eine Millisekunde ist, überschreitet demgemäß die Differenz D den Schwellenwert Dt. Wenn dies der Fall ist, wird erkannt, daß die Computer nicht gleichmäßig ausgelastet sind, wodurch ein zu verschiebendes Programm ausgewählt wird (Schritt 355). Bei diesem Ausführungsbeispiel ist angenommen, daß aus den ausführbaren Programmen, die der Zeitgrenze pro Einheitsverarbeitungs menge von t Millisekunden oder weniger entsprechen, zu diesem



Zweck dasjenige Programm ausgewählt wird, das das letzte in der Prioritätsschlange ist. Wie in Fig. 30 dargestellt, sind die Prioritätswerte, die dem Zeitgrenzbereich von 0,05 bis 1 Millisekunde des Computers 100 entsprechen, die Werte 2—4, und das Programm K ist als dasjenige vorgesehen, das als nächstes innerhalb der Prioritätsschlange auszuführen ist (Programmstruktur 597). So wird das Programm K als zu verschiebendes Programm ausgewählt.

Wenn das zu verschiebende Programm einmal bestimmt ist, fordert die Programmverschiebe-Bestimmungseinrichtung des Computers, in dem das zu verschiebende Programm liegt, die zugehörige Programmausführungs- und Programmverschiebeeinrichtung dazu auf, den Vorgang auszuführen, wie er zu einer Programmverschiebung gehört. Bei diesem Beispiel ist die Programmausführungs- und Programmverschiebeeinrichtung 290 des Computers 100 mit diesem Vorgang beauftragt. Sie fordert die Computerauswahleinrichtung 298 dazu auf, einen Zielcomputer auszuwählen, an den das ausgewählte Programm zu verschieben ist (Schritt 356). Die Computerauswahleinrichtung 298 kommuniziert mit den Computerauswahleinrichtungen in den anderen Computern, um den Zielcomputer zu bestimmen. Bei diesem Ausführungsbeispiel ist angenommen, daß der Computer mit dem kleinsten Auslastungs/Funktionsvermögen-Wert als Zielcomputer ausgewählt wird. Daher wird bei diesem Beispiel der Computer 101, der den kleinsten Auslastungs/Funktionsvermögen-Wert hat, wenn  $t$  eine Millisekunde ist, als Zielcomputer ausgewählt.

Wenn das Ziel einmal bestimmt ist, führt die Programmausführungs- und Programmverschiebeeinrichtung den Programmverschiebevorgang aus (Schritt 357). Bei diesem Beispiel führt die Programmausführungs- und Verschiebeeinrichtung 290 des Computers 100 die Programmverschiebeeinrichtung in Übereinstimmung mit der Programmausführungs- und Programmverschiebeeinrichtung 291 des Zielcomputers 101 aus. Nachdem das Programm auf den Computer 101 verschoben wurde, wird der durch diesen Verschiebevorgang verbesserte Auslastungszustand in den Computerauslastungs-Informationstabellen 370, 371 wiedergegeben, wie in Fig. 34 dargestellt. Wie es aus einem Vergleich der Fig. 33 mit Fig. 34 erkennbar ist, ist in den Computerauslastungs-Informationstabellen 370, 371 die Anzahl ausführbarer Programme im Bereich der Zeitgrenze pro Einheitsverarbeitungs menge von 0,05 bis 1 Millisekunde von acht auf sieben für den Computer 100 verringert, dagegen von eins auf zwei für den Computer 101 erhöht.

Nachdem das Programm wie vorstehend beschrieben verschoben wurde, kehrt der Ablauf erneut zum Schritt 352 zurück, um die Auslastungs/Funktionsvermögen-Werte für die jeweiligen Computer zu berechnen. Für diesen Fall sind die berechneten Auslastungswerte und Auslastungs/Funktionsvermögen-Werte diejenigen, wie sie in den Tabellen 380, 381, 382, 383 in Fig. 34 dargestellt sind. Obwohl die Information für die mit 102 und den folgenden Bezugswerten bezeichneten Computer aus Fig. 34 weggelassen ist, gilt, daß die Differenz  $D$  immer noch den Schwellenwert  $Dt$  übersteigt, wenn  $t$  eine Millisekunde ist, weswegen erneut ein Programmverschiebevorgang ausgeführt wird, wie vorstehend beschrieben wurde. Wenn die Differenz  $D$  den Wert  $Dt$  hat oder kleiner ist, geht der Ablauf zum Schritt 358 weiter.

Durch Wiederholen des vorstehend genannten Ver-

arbeitungsablaufs, wie er in Fig. 33 dargestellt ist, bis der Maximalwert des Zeitgrenzbereichs auf den Wert  $t$  gesetzt ist, wird Programmverschiebung ausgeführt, um gleichmäßige Auslastungsverteilung für die zum verteilten Computersystem gehörigen Computer abhängig von deren Funktionsvermögen zu erzielen, was für alle Zeitgrenzbereiche erfolgt.

Es wurde das System gemäß dem zehnten Ausführungsbeispiel der Erfindung beschrieben. Dieses System des zehnten Ausführungsbeispiels hat den Vorteil, daß die Auslastung im System, d. h. durch diese auszuführenden Benutzerprogramme, gleichmäßig auf alle zum System gehörenden Computer abhängig von deren jeweiligem Funktionsvermögen verteilt werden kann, so daß die Computermittel wirkungsvoll genutzt werden können. Ferner beeinflussen Schwankungen der Funktionsvermögen der Computer eine gleichmäßige Bewertung der Auslastung nicht, da die Gleichmäßigkeit der Auslastungsverteilung unter Berücksichtigung sowohl der Auslastungswerte als auch der Funktionsvermögen der jeweiligen Computer bewertet wird.

Auch ist das System dieses Ausführungsbeispiels dahingehend von Vorteil, daß Benutzerprogramme gleichmäßig auf Computer abhängig von deren jeweiligem Funktionsvermögen für jeden Zeitgrenzbereich verteilt werden können, so daß Echtzeit-Verarbeitung auf Grundlage einer gleichmäßigen Zeitgrenze im gesamten verteilten Computersystem erzielt werden kann. Insbesondere ist es möglich, da die Auslastung im gesamten System unter Berücksichtigung der jeweiligen Programme auferlegten Zeitgrenzen verteilt wird, zu verhindern, daß eine Anzahl von Programmen mit kurzer Zeitgrenze auf einen einzelnen Computer konzentriert wird, wodurch eine strengere Einhaltung von Echtzeiteigenschaften gewährleistet ist.

Während das in den Fig. 30—34 dargestellte System über Prioritätswert-Umsetzeinrichtungen 572, 573, Computerauswahleinrichtungen 298, 299 und Computerinformationsbereiche 330, 331 verfügt, die verteilt in den jeweiligen Computern 100, 101 angeordnet sind, können diese Einrichtungen für eine konzentrierte Verwaltung in einem einzelnen Computer angeordnet sein.

Das unter Bezugnahme auf die Fig. 17—21 beschriebene System ist ein solches, daß mehrere Programme, die denselben Vorgang ausführen, auf mehreren Computern ausgeführt werden, wodurch eine Verarbeitungsanforderung ausgegeben wird, die einen Dringlichkeitswert spezifiziert, um aus den denselben Vorgang ausführenden Programmen ein solches auszuwählen, das den Prioritätswert aufweist, wie er vom anfordernden Programm gewünscht wird. Es wird nun ein elftes Ausführungsbeispiel der Erfindung erläutert, das dazu dient, diese Programmauswahl unter Verwendung der Zeitgrenze statt der Dringlichkeit auszuführen. Fig. 35 zeigt ein verteiltes Computersystem gemäß dem elften Ausführungsbeispiel, das die Funktion hat, daß es Verarbeitungsanforderungen unter Verwendung der Zeitgrenze verteilt. Genauer gesagt, verwendet das System von Fig. 35 die Zeitgrenze anstelle der Dringlichkeit, wie sie vom System von Fig. 17 verwendet wird. In Fig. 35 sind die Elemente 100—103, 110 sowie 140—143 identisch mit denjenigen Elementen, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind. Einzelheiten zur Hardware der Computer 100, 101 sind weggelassen. Die Elemente 430—443, 450, 451, 454—458, 460—462 sowie 470—479 in Fig. 35 sind identisch mit denjenigen Elementen, die in Fig. 17 mit denselben Bezugswerten gekennzeichnet sind. Eine Prioritätswert-



Umsetzeinrichtung 720 und eine Prioritätswert-Entsprechungstabelle 730 sind dergestalt, daß die Prioritätswert-Umsetzeinrichtungen 572, 573 und die Prioritätswert-Entsprechungstabellen 580, 581 in Fig. 22 so modifiziert sind, daß sie einen einzigen Ablauf und eine einzige Tabelle bilden, damit die Verwaltung mit diesem einzigen Ablauf und der einzigen Datenstruktur möglich ist. Die Prioritätswert-Umsetzeinrichtung 720 kann für jeden Computer einen örtlichen Prioritätswert erstellen, der einer spezifizierten Zeitgrenze entspricht, und zwar unter Verwendung der in Fig. 24 dargestellten Umsetzeinrichtung 620.

Eine Programmauswahleinrichtung 700 wählt ein Programm aus, wie es für einen anfragenden Computer optimal ist, der eine Zeitgrenze vorgibt. Eine Programmort-Datenbank 710 ist als Verwaltungsdatenstruktur für diese Auswahl vorhanden. Die Programmort-Datenbank 710 verwaltet Programmort-Informationstabellen 470—479 zum Einspeichern von Kennungen von Computern, auf denen Programme, die denselben Vorgang ausführen, laufen, und sie speichern die Prioritätswerte der Programme. Die Programmort-Informationstabellen 470—479 werden als Programmlisten 711—716 für jeweilige Sätze von Programmen, die denselben Vorgang ausführen, verwaltet. Bei diesem Ausführungsbeispiel ist angenommen, daß die Programmlisten 711—716 selbst den Verarbeitungsmenge-Datenwert 600, wie er in Fig. 22 dargestellt ist, für die zugehörigen Programme speichert. Auf diese Weise kann eine Zeitgrenze pro Einheitsverarbeitungs-menge aus einer spezifizierten Zeitgrenze und einem Verarbeitungsmenge-Datenwert 600, wie von einer Programmliste 711—716 spezifiziert, hergeleitet werden. Alternativ kann der Verarbeitungsmenge-Datenwert 600 in der Programmstruktur abgespeichert sein, so daß die Prioritätswert-Umsetzeinrichtung 720 den Verarbeitungsmenge-Datenwert 600 liest, um eine Zeitgrenze pro Einheitsverarbeitungs-menge herzuleiten, wie im Fall des in Fig. 22 dargestellten Systems. Wenn die Prioritätswerte der in der Programmauswahleinrichtung 700 registrierten Programme in irgendeinem Computer verändert werden, wenn ein Programm aus einem zugehörigen Computer gelöscht wird oder wenn ein zu registrierendes Programm neu gestartet wird, wird die Programmauswahleinrichtung 700 über die Kennung des Computers informiert, auf dem das betreffende Programm läuft, wie auch über den Dringlichkeitswert des Programms, um den Inhalt der Programmort-Datenbank 710 zu modifizieren.

Nachfolgend erfolgt eine Erläuterung für ein Verfahren zum Auswählen eines Zielprogramms, an das eine Verarbeitungsanforderung geliefert wird, abhängig von der Zeitgrenze, unter Verwendung der Prioritätswert-Umsetzeinrichtung 720 und der Programmauswahleinrichtung 700. Fig. 36 zeigt ein Beispiel für den Programmablauf zum Auswählen eines Programms. Bei diesem Ausführungsbeispiel ist angenommen, daß ein Benutzerprogramm 440 eine Verarbeitungsanforderung an ein Programm in einer Programmgruppe D ausgibt (Benutzerprogramm 435 oder 436). Das Benutzerprogramm 440 befragt zunächst die Programmauswahleinrichtung 700 über den Ort eines Programms, das zur Programmgruppe D gehört und das einer Zeitgrenze von fünf Millisekunden genügt (Verbindung 750). Die Programmauswahleinrichtung 700 ruft den Verarbeitungsmenge-Datenwert 600 für das Programm in der Gruppe D aus der Programmliste 714 ab (Schritt 751). Hier sei angenommen, daß die Verarbeitungsmenge des

Programms in der Gruppe D 12.000 Anweisungen ist. Aus der vorgegebenen Zeitgrenze und dem Verarbeitungsmenge-Datenwert 600 wird die Zeitgrenze pro Einheitsverarbeitungs-menge zu 0,039 Millisekunden berechnet ( $5.00/128 = 0,039$ ) (Berechnung 752). Die Programmauswahleinrichtung 700 informiert die Prioritätswert-Umsetzeinrichtung 720 über das durch die Berechnung 752 gewonnene Ergebnis und fordert diese dazu auf, die Zeitgrenze pro Einheitsverarbeitungs-menge in einen Prioritätswert umzusetzen (Verbindung 753). Die Prioritätswert-Umsetzeinrichtung 720 führt den in Fig. 24 veranschaulichten Ablauf 620 aus, um örtliche Prioritätswerte für die jeweiligen Computer zu bestimmen, die der Zeitgrenze von 0,039 Millisekunden entsprechen, und sie informiert die Programmauswahleinrichtung 700 in Form der Prioritätswertliste 492 über die bestimmten Prioritätswerte (Verbindung 754). Die Prioritätswertliste 492 gibt die örtlichen Prioritätswerte für die jeweiligen Computer an, die der spezifizierten Zeitgrenze genügen. Bei diesem Ausführungsbeispiel empfängt die Programmauswahleinrichtung 700 daraufhin eine Prioritätswertliste, die angibt, daß die Zeitgrenze von 0,039 Millisekunden dem Prioritätswert 1 im Computer 100 und dem Prioritätswert 0 im Computer 101 entspricht, was über die Verbindung 754 erfolgt. Die Programmauswahleinrichtung 700 vergleicht den Inhalt der Programmort-Datenbank 710 mit dem Inhalt der Prioritätswertliste 492 zum Bestimmen eines Zielprogramms, an das die Verarbeitungsanforderung zu liefern ist. Der Verarbeitungsablauf 500 in Fig. 19 kann dazu verwendet werden, die Programmliste 711—716 mit der Prioritätswertliste 492 zu vergleichen, um ein Programm mit einem Prioritätswert herauszufinden, der im spezifizierten Prioritätswertebereich liegt. Beim in Fig. 36 dargestellten Ausführungsbeispiel enthält die Programmliste 714 für die Programmgruppe D nur die Programmortinformation 476 (entsprechend dem Benutzerprogramm 436), die dem Prioritätswertebereich entspricht, wie er durch die Prioritätswertliste 492 spezifiziert wird. Daher bestimmt die Programmauswahleinrichtung 700, daß das im Computer 101 liegende Benutzerprogramm 436 als Ziel geeignet ist, an das die Verarbeitungsanforderung ausgegeben werden kann, und sie informiert das Benutzerprogramm 440 über dieses Bestimmungsergebnis (Verbindung 755). Das Benutzerprogramm 440 gibt dann die Verarbeitungsanforderung an das Benutzerprogramm 436 aus, das über die Verbindung 755 aufgerufen wurde. Alternativ ist es möglich, ein Kommunikation-Zwischenschaltungsverfahren zu verwenden, bei dem die Programmauswahleinrichtung 700 die Information 750 vom Benutzerprogramm 440 direkt an das Benutzerprogramm 436 überträgt (Verbindung 757).

Das verteilte Rechnersystem des elften Ausführungsbeispiels ermöglicht es, daß mehrere Programme mit verschiedenen Prioritätswerten, die denselben Vorgang ausführen, verteilt in zum System gehörenden Computern liegen können, wodurch eine Verteilung von Verarbeitungsanforderungen oder der Auslastung abhängig von einer vorgegebenen Zeitgrenze erzielt wird. Insbesondere ist eine gute Wirkung dieses Ausführungsbeispiels zu erwarten, da selbst Verarbeitungsanforderungen, die nach demselben Inhalt der Verarbeitung fragen, häufig mit verschiedenen Zeitgrenzen versehen werden, abhängig von den anfordernden Programmen.

Weiter kann gemäß dem elften Ausführungsbeispiel dadurch, daß für die mehreren Programme, die denselben Vorgang ausführen, verschiedene Prioritätswerte



eingestellt werden, derselbe Verarbeitungsinhalt mit verschiedenen Zeitgrenzen angefordert werden, ohne daß eine Prioritätswert-Umsetzeinrichtung erforderlich ist, wodurch es ermöglicht ist, Echtzeiteigenschaften im selben Ausmaß wie beim achten Ausführungsbeispiel zu gewährleisten.

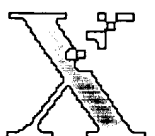
Fig. 37 zeigt ein zwölftes Ausführungsbeispiel, bei dem es sich um ein anderes verteiltes Computersystem mit Programmauswahlfunktion, wie vorstehend beschrieben, handelt. Das verteilte Computersystem von Fig. 37 beinhaltet eine Kombination aus Prioritätswert-Umsetzeinrichtungen 572, 573, die getrennt in jeweiligen Computern vorhanden sind, und einer Programmauswahleinrichtung 760, die Kennungen von Computern, auf denen denselben Vorgang ausführende Programme laufen, und den Programmen zugeordnete Zeitgrenzen speichert, um ein Ziel zu bestimmen, daß eine Verarbeitungsanforderung auszugeben ist. In Fig. 37 sind die Elemente 100—103, 110 sowie 140—143 mit denjenigen Elementen identisch, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind. Einzelheiten zur Hardware der Computer 100, 101 sind weggelassen. Die Benutzerprogramme 430—442 sowie die Programmstrukturen 450—462 haben die gleichen Inhalte wie in Fig. 17. Die Programmstrukturen, die den Benutzerprogrammen 432—433 sowie 439 entsprechen, sind weggelassen. Die Prioritätswert-Umsetzeinrichtungen 572, 573, die Prioritätswert-Entsprechungstabellen 580, 581 und der Verarbeitungsmenge-Datenwert 600 entsprechen jeweils denselben Abläufen und Datenstrukturen wie in Fig. 22.

Die Programmauswahleinrichtung 700 verwaltet Computer, auf denen diese Programme laufen, wie auch die Zeitgrenzen, um ein Programm mit optimaler Zeitgrenze zu bestimmen. Eine Programmort-Datenbank 770 ist zu diesem Zweck als Verwaltungsdatenstruktur vorhanden. Die Programmort-Datenbank 770 verwaltet Programmort-Informationstabellen 780—789 zum Ein Speichern von Kennungen von Computern, auf denen denselben Vorgang ausführende Programme laufen, sowie deren Zeitgrenzen (es ist zu beachten, daß die Zeitgrenze in diesem Fall eher eine Verarbeitungsvorausagezeit repräsentiert). Die Programmort-Informationstabellen 780—789 bilden Programmlisten 771—776, so daß jeder Satz von Programmen, die denselben Vorgang ausführen, eine Liste bilden. Während die Programmort-Informationstabellen 470—479 in Fig. 35 Kennungen von Computer, auf denen denselben Vorgang ausführende Programme laufen und deren Prioritätswerte speichern, speichern die Programmort-Informationstabellen 780—789 in Fig. 37 Zeitgrenzen anstelle von Prioritätswerten. Die Zeitgrenze kann dadurch berechnet werden, daß die Zeitgrenze pro Einheitsverarbeitungs menge, wie sie dem Prioritätswert eines Programms entspricht, mit der Verarbeitungsmenge des Programms multipliziert wird. Z.B. läuft das Benutzerprogramm 436 (in der Programmgruppe D) auf dem Computer 101 mit dem Prioritätswert 0. Wenn angenommen wird, daß die Prioritätsmenge des Programms in der Programmgruppe D 128.000 Anweisungen umfaßt, wird diese Zeitgrenze zu 4,35 Millisekunden berechnet ( $0,034 \times 128 = 4,35$ ). Dieser Wert wird in den Programmort-Informationsbereich 786 eingespeichert (Ortsinformationsbereich, der dem Benutzerprogramm 436 entspricht). Wenn der Prioritätswert eines in der Programmauswahleinrichtung 760 registrierten Programms in irgendeinem Computer geändert wird, wenn ein Programm aus irgendeinem Computer gelöscht wird

oder wenn ein zu registrierendes Programm neu gestartet wird, wird die Programmauswahleinrichtung 760 über den zu diesem Programm gehörenden Computer und den Prioritätswert des Programms informiert, um den Inhalt der Programmlisten zu modifizieren. Auch dann, wenn die Entsprechungsbeziehung zwischen dem Prioritätswert und der Zeitgrenze für irgendeinen Computer geändert wird, informieren die Prioritätswert-Umsetzeinrichtungen 572, 573 die Programmauswahleinrichtung 760 über derartige Änderungen der Entsprechungsbeziehung, um die zugehörigen Zeitgrenzen zu ändern, die in den Programmort-Informationsbereichen 780—789 abgespeichert sind.

Fig. 37 veranschaulicht auch ein Verfahren zum Auswählen eines Zielprogramms, an das eine Verarbeitungsanforderung ausgegeben wird, abhängig von der Zeitgrenze, und zwar unter Verwendung der Prioritätswertumsetzeinrichtungen 572, 573 sowie der Programmauswahleinrichtung 760. Auf ähnliche Weise wie bei der Erläuterung zum Verarbeitungsablauf in Fig. 36 ist es auch bei diesem Ausführungsbeispiel angenommen, daß ein Benutzerprogramm 440 eine Anforderungsanweisung an ein Programm in einer Programmgruppe D (Benutzerprogramm 435 oder 436) ausgibt. Das Benutzerprogramm 440 befragt zunächst die Programmauswahleinrichtung 760 hinsichtlich des Orts eines Programms, das zur Programmgruppe D gehört und eine Zeitgrenze von fünf Millisekunden einhält (Verbindung 790). Da die Programmauswahleinrichtung 760 die Kennungen von Computern, auf denen jeweils gleiche Programme laufen, und deren Zeitgrenzen erkannt hat, kann sie dabei den Ort eines Programms herausfinden, das den Bedingungen genügt, daß das Programm zur Programmgruppe D gehört und seine Zeitgrenze fünf Millisekunden entspricht oder kürzer ist. Bei diesem Ausführungsbeispiel genügt innerhalb der zwei Benutzerprogramme in der Programmgruppe D nur die Programmortinformation 786 (entsprechend dem Benutzerprogramm 436) den vorstehend genannten Bedingungen. Daher bestimmt die Programmauswahleinrichtung 700, daß das im Computer 101 liegende Benutzerprogramm 436 als Ziel geeignet ist, an das die Verarbeitungsanforderung ausgegeben werden kann, und sie informiert das Benutzerprogramm 440 über diese Bestimmung (Verbindung 791). Das Benutzerprogramm 440 gibt dann die Verarbeitungsanforderung an das Benutzerprogramm 436 aus, das über die Verbindung 791 aufgefunden wurde. Alternativ ist es möglich, ein Kommunikations-Zwischenschaltungsverfahren zu verwenden, bei dem die Programmauswahleinrichtung 760 Information 790 vom Benutzerprogramm 440 direkt an das Benutzerprogramm 436 liefert (Verbindung 793).

Das System gemäß dem zwölften Ausführungsbeispiel hat dieselbe Wirkung wie das des elften Ausführungsbeispiels und ist darüber hinaus dahingehend von Vorteil, daß die Programmauswahl schneller als bei dem in den Fig. 35, 36 dargestellten Systemen erfolgen kann, da die Programmauswahleinrichtung 760 nicht die Prioritätswert-Umsetzeinrichtungen 572, 573 dazu anweist, eine Zeitgrenze in einen entsprechenden Prioritätswert umzusetzen. Es ist jedoch zu beachten, daß, da die Programmort-Informationsbereiche 780—789 Zeitgrenzen enthalten, diese Zeitgrenzen in diesen Programmort-Informationsbereichen unter Umständen geändert werden müssen, wenn Änderungen in der Auslastung irgendeines Computers bewirken, daß sich die Entsprechungsbeziehung zwischen der Zeitgrenze und dem Prioritätswert ändert. Daher ist das in den Fig. 35, 36





dargestellte System dann von Vorteil, wenn sich die Auslastungen der jeweiligen Computer zeitlich stark ändern.

Abweichend von dem in den Fig. 35, 36 dargestellten System und dem in Fig. 37 dargestellten System ist es auch möglich, ein System aufzubauen, das eine Kombination aus der Programmauswahleinrichtung 760 und der Prioritätswert-Umsetzeinrichtung 720 enthält, oder ein System, das die Programmauswahleinrichtung 700 und die Prioritätswert-Umsetzeinrichtungen 572, 573 enthält. Ferner sind zwar die Prioritätswert-Umsetzeinrichtungen 146, 147 für die in den Fig. 35—37 dargestellten Systeme nicht beschrieben, da diese Systeme nicht davon ausgehen, daß sie ein Programm mit verändertem Prioritätswert starten müssen, jedoch können diese Einrichtungen hinzugefügt sein, damit das System die Prioritätswerte ändern kann.

Schließlich zeigt Fig. 38 ein System gemäß einem dreizehnten Ausführungsbeispiel der Erfindung, das eine Zeitgrenze dazu verwendet, um ein ein Programm ausgegebene Verarbeitungsanforderung auf ein anderes Programm zu verlegen, die denselben Vorgang wie das angeforderte Programm ausführt, ähnlich wie dies beim verteilten Rechnersystem in Fig. 21 der Fall ist. In Fig. 38 sind die Elemente 100—103, 110 sowie 140—143 identisch mit denjenigen Elementen, die in Fig. 1 mit denselben Bezugswerten gekennzeichnet sind. Einzelheiten zur Hardware der Computer 100, 101 sind weggelassen. Prioritätswert-Umsetzeinrichtungen 572, 573 sowie ein Verarbeitungsmenge-Datenwert 600 entsprechen jeweils demselben Datenablauf sowie Datenstruktur wie in Fig. 22. Benutzerprogramme 430—443 sowie Programmstrukturen 450—462 haben jeweils denselben Inhalt wie Fig. 17. Programmstrukturen, die den Benutzerprogrammen 432, 433, 439 entsprechen, sind weggelassen. Auch sind die Elemente 760, 760—776 sowie 780—789 in Fig. 38 identisch mit denjenigen Elementen, die in Fig. 37 mit denselben Bezugswerten bezeichnet sind. Die Verarbeitungsverlegeeinrichtung 550, die Schlange 551 sowie die Verarbeitungsanforderungen 552, 553 sind identisch mit den entsprechenden Einheiten in Fig. 21.

Um das Konzept der Zeitgrenze zu verwenden, kann eine Bedingung zum Verlegen einer angeforderten Verarbeitung auf ein anderes Programm die folgenden zwei repräsentativen Beispiele umfassen:

- (1) Eine Anforderung für eine neue Verarbeitung wird an ein Programm ausgegeben, das gerade schon eine andere Verarbeitung ausführt; und
- (2) es wird erkannt, daß eine aktuell ausgeführte Verarbeitung nicht rechtzeitig beendet wird.

Im ersteren Fall ermöglichen es die Verarbeitungsverlegeeinrichtung 550 und die Schlange 551, daß eine Verarbeitungsanforderung auf ein anderes Programm verlegt werden kann, ähnlich wie bei der in Fig. 21 dargestellten Situation. Jedoch unterscheidet sich dieses Ausführungsbeispiel dahingehend, daß die Verarbeitungsverlegeeinrichtung 550 eine Zeitgrenze für eine Programmauswahleinrichtung 760 spezifiziert.

Im letztgenannten Fall kann die Verarbeitungsverlegung dadurch erzielt werden, daß jedes Programm mit einer Verarbeitungszeit-Voraussageeinrichtung 800 und einer Zeitgrenze-Einstelleinrichtung 802 versehen wird. Die Verarbeitungszeit-Voraussageeinrichtung 800 sagt eine Zeit voraus, wie sie dazu erforderlich ist, eine aktuell ausgeführte Verarbeitung abzuschließen. Die

Voraussage kann z. B. dadurch erfolgen, daß für jedes Programm eine Verarbeitungsrestmenge 801 abgespeichert wird. Wenn z. B. angenommen wird, daß ein Benutzerprogramm 438 eine Verarbeitungsrestmenge 801 mit dem Wert 60 aufweist, kann, da die Zeitgrenze pro Einheitsverarbeitungsanforderung des Benutzerprogramms 438 (Prioritätswert 1 im Computer 101) 0,074 Millisekunden ist, die zum Abschließen des Programms 438 erforderliche Zeit zu 4,44 Millisekunden ( $0,074 \times 60 = 4,44$ ) abgeschätzt werden. Die Zeitgrenze-Einstelleinrichtung 802 berechnet die neue Zeitgrenze, wenn das zugehörige Programm eine Verarbeitungsanforderung neu an ein anderes Programm ausgibt. Diese Einstellung kann dadurch bewerkstelligt werden, daß die Zeit 803 abgespeichert wird, die ab dem Zeitpunkt verstrichen ist, zu dem die Verarbeitungsanforderung empfangen wurde, oder dergleichen.

Der durch Verbindungen 810—816 veranschaulichte Ablauf repräsentiert, wie eine Verarbeitungsanforderung von einem Programm auf ein anderes verlegt wird, wenn eine solche Verarbeitungsanforderung zu verlegen ist. Das Ausführungsbeispiel von Fig. 38 zeigt, daß das Benutzerprogramm 440 eine Verarbeitungsanforderung an das Benutzerprogramm 438 (Programm in einer Programmgruppe F) mit einer Zeitgrenze von fünf Millisekunden ausgibt (Verbindung 810), und diese Verarbeitungsanforderung wird vom Benutzerprogramm 438 auf ein anderes Programm verlegt. Es ist angenommen, daß die Verarbeitungsrestmenge 60.000 Anweisungen ist, und zwar 1,8 Millisekunden nach dem Zeitpunkt, zu dem die Verarbeitungsanforderung 552 ausgegeben wurde. Die Verarbeitungszeit-Voraussageeinrichtung 800 empfängt von der Prioritätswert-Umsetzeinrichtung 573 eine Zeitgrenze pro Einheitsverarbeitungsanforderung von 0,074 Millisekunden, was dem Prioritätswert 1 des Programms 440 entspricht (Verbindung 811), und aus der obengenannten Berechnung wird eine Verarbeitungsvoraussagezeit von 4,44 Millisekunden hergeleitet (Verbindung 812). Die Zeitgrenze-Einstelleinrichtung 802 berechnet ausgehend von der Zeitgrenze von fünf Millisekunden und der verstrichenen Zeit von 1,8 Millisekunden eine neue Zeitgrenze von 3,2 Millisekunden (Verbindung 813). Da vorausgesagt ist, daß die restliche Verarbeitung nicht innerhalb der neuen Zeitgrenze abgeschlossen werden kann, wird diese Verarbeitungsanforderung an ein anderes Programm verlegt. Für diese Verlegung befragt die Verarbeitungsverlegeeinrichtung 550 die Programmauswahleinrichtung 760 unter Angabe der neu berechneten Zeitgrenze von 3,2 Millisekunden, ob in der Programmgruppe F ein Programm vorhanden ist, das diese Zeitgrenze erfüllt (Verbindung 814). Die Programmauswahleinrichtung 760 durchsucht eine Programmliste 776 für die Programmgruppe F nach Programmortsinformation für ein Programm, das der Zeitgrenze von 3,2 Millisekunden genügt. Bei dem in Fig. 38 dargestellten Beispiel kann die Programmauswahleinrichtung 760 die Programmortsinformation 789 auffinden (entsprechend einem Benutzerprogramm 439 im Computer 103). Die Programmauswahleinrichtung 760 informiert die Verarbeitungsverlegeeinrichtung 550 über den Ort des Benutzerprogramms 439, das der Programmortsinformation 789 entspricht (Verbindung 815).

Die Verarbeitungsverlegeeinrichtung 550 fordert dann, wenn sie diese Information empfängt, das Benutzerprogramm 439 dazu auf, die Verarbeitung 552 auszuführen, für die erkannt wurde, daß sie nicht innerhalb der Zeitgrenze abgeschlossen wurde (Verbindung 816).

Durch das dreizehnte Ausführungsbeispiel kann eine



an ein Programm ausgegebene Verarbeitungsanforderung auf ein anderes Programm verlegt werden, das dieselbe Zeitgrenze wie das angeforderte Programm hat, um zu verhindern, daß eine Anzahl von Verarbeitungsanforderungen auf einen einzelnen Computer konzentriert wird, was es ermöglicht, die Verarbeitungsanforderungen auf mehrere Programme zu verteilen. Da eine neu angeforderte Verarbeitung sofort ausgeführt werden kann, ohne auf den Abschluß einer aktuell ausgeführten Verarbeitung warten zu müssen, sind ferner verbesserte Echtzeiteigenschaften sichergestellt.

Gemäß diesem Ausführungsbeispiel kann eine an ein Programm ausgegebene Verarbeitungsanforderung auf ein anderes Programm verlegt werden, das die Verarbeitung schneller als das angeforderte Programm abschließen kann, und zwar wenn erkannt wird, daß das angeforderte Programm die Verarbeitung nicht rechtzeitig abschließen kann, wodurch strenger sichergestellt ist, daß die erforderliche Verarbeitung innerhalb der Zeitgrenze abgeschlossen wird. Die Verlegung einer Verarbeitungsanforderung soll in diesem Fall gewährleisten, daß die Zeitgrenze selbst dann erfüllt wird, wenn irgendein unerwarteter Zwischenfall dazu führt, daß die Ausführung eines Programms verzögert wird.

Während das in Fig. 38 dargestellte System über eine Verarbeitungsverlegeeinrichtung 550, eine Schlange 551, eine Verarbeitungszeit-Voraussageeinrichtung 800 und eine Zeitgrenze-Einstelleinrichtung 802 in jedem Programm verfügt, können diese Elemente außerhalb der Programme angeordnet sein, so daß sie von mehreren Programmen gemeinsam genutzt werden können. Zusammengefaßt gesagt, besteht das verteilte Computersystem dieses Ausführungsbeispiel aus einer Kombination der Programmauswahleinrichtung 760 und der Prioritätswert-Umsetzeinrichtungen 572, 573, wie in Fig. 37 dargestellt, wobei zusätzlich die Verarbeitungsverlegeeinrichtung 550, die Schlange 551, die Verarbeitungszeit-Voraussageeinrichtung 800 und die Zeitgrenze-Einstelleinrichtung 802 vorhanden sind. Alternativ können diese zusätzlichen Vorgänge und Datenstrukturen zum in Fig. 35 dargestellten verteilten Rechnersystem hinzugefügt sein, das eine Kombination aus einer Programmauswahleinrichtung 700 und einer Prioritätswert-Umsetzeinrichtung 720 beinhaltet, um ähnliche Vorgänge auszuführen.

Ausgehend vom vorstehend genannten dreizehnten Ausführungsbeispiel kann eine alternative Konfiguration erzeugt werden. Genauer gesagt, verlegt das System des dreizehnten Ausführungsbeispiels eine Verarbeitungsanforderung auf ein anderes Programm, wenn festgestellt wird, daß das angeforderte Programm eine auferlegte Zeitgrenze nicht erfüllen kann, jedoch ist es auch möglich, einem Programm, das zum Ausführen der Verarbeitung angefordert wird, einen höheren Prioritätswert zu geben, anstatt die Verarbeitung zu verlegen, so daß die aktuell ausgeführte Verarbeitung schneller abgeschlossen wird. Bei dieser Alternative wird bei dem in Fig. 38 dargestellten System aus der neuen Zeitgrenze von 3,2 Millisekunden und der Verarbeitungsrestmenge von 60.000 Anweisungen eine Zeitgrenze pro Einheitsverarbeitungs Menge von 0,053 Millisekunden berechnet ( $3,2/60 = 0,053$ ), und der Prioritätswert des Benutzerprogramms 438 wird auf den Prioritätswert 0 angehoben, um der berechneten Zeitgrenze pro Einheitsverarbeitungs Menge zu genügen, wodurch das Benutzerprogramm 438 die Verarbeitung mit dem erhöhten Prioritätswert fortsetzt.

Es ist zu beachten, daß die Verbindungszeit-Voraus-

sageeinrichtungen 670, 671 in Fig. 29 in die Systeme der Fig. 30—38 eingeführt werden können, um die Zeitgrenze genauer zu bestimmen.

Wie es aus der Beschreibung zum ersten bis dreizehnten Ausführungsbeispiel ersichtlich ist, erzielt die Erfindung gleichmäßige Verwaltung örtlich definierter Prioritätswerte in jeweiligen Computern, die zu einem verteilten Computersystem gehören, unter Verwendung eines Dringlichkeitswerts oder einer Zeitgrenze, um eine strengere Einhaltung von Echtzeiteigenschaften zu gewährleisten.

Beim erfindungsgemäßen System, das den Prioritätswert mittels eines Dringlichkeitswerts verwaltet, sind eine Prioritätswert-Umsetzeinrichtung und eine Prioritätswert-Änderungseinrichtung dazu verwendet, sicherzustellen, daß Verarbeitungen mit demselben Dringlichkeitswert in allen Computern, auch wenn diese unterschiedliches Funktionsvermögen und unterschiedliche Auslastung aufweisen, im wesentlichen mit derselben Geschwindigkeit ausgeführt werden.

Auch verhindert die beim erfindungsgemäßen System verwendete Computerauswahleinrichtung, daß eine Anzahl von Verarbeitungsanforderungen mit hohem Dringlichkeitswert auf einen einzelnen Computer konzentriert wird, wodurch es ermöglicht ist, Echtzeiteigenschaften in den jeweiligen Computern strenger einzuhalten.

Ferner werden durch die Programmauswahleinrichtung mehrere Programme mit verschiedenen Dringlichkeitswerten bereitgestellt, so daß eine Verarbeitungsanforderung an das Programm mit dem geeignetsten Dringlichkeitswert unter ihnen ausgegeben wird, selbst wenn keine Prioritätswert-Änderungseinrichtung verändert wird, mit dem Ergebnis, daß die Verarbeitung mit gewünschter Geschwindigkeit ausgeführt werden kann.

Ferner kann unter Verwendung einer Verarbeitungsverlegeeinrichtung eine Verarbeitung, wie sie von einem aktuell laufenden Programm gefordert wird, an ein anderes Programm verlegt werden und von diesem ausgeführt werden, das die Ausführung mit im wesentlichen derselben Geschwindigkeit vornehmen kann, ohne daß auf den Abschluß der gerade ausgeführten Verarbeitung gewartet werden muß, wodurch eine strengere Einhaltung von Echtzeiteigenschaften gewährleistet ist.

Beim erfindungsgemäßen System, das den Prioritätswert mittels der Zeitgrenze verwaltet, werden eine Prioritätswert-Umsetzeinrichtung und eine Prioritätswert-Änderungseinrichtung dazu verwendet, eine Verarbeitung abhängig vom örtlichen Prioritätswert in einem speziellen Computer abhängig von einer spezifizierten Zeitgrenze auszuführen. Dies ermöglicht es, daß eine Verarbeitung mit einer strengeren Zeitgrenze vor einer anderen Verarbeitung mit einer weniger strengen Zeitgrenze ausgeführt wird, so daß die Echtzeiteigenschaften für jede Verarbeitung im verteilten Computersystem strenger gewährleistet werden können. Auch dann, wenn eine Verarbeitung mit relativ langer Zeitgrenze unbearbeitet bleibt, um bevorzugt andere Verarbeitungen mit kurzer Zeitgrenze auszuführen, nähert sich die verstrichene Zeitgrenze der auferlegten Zeitgrenze an. In diesem Fall kann der Prioritätswert für die Verarbeitung abhängig von der verkürzten Zeitgrenze auf einen höheren Wert korrigiert werden, und die Verarbeitung wird mit diesem korrigierten Prioritätswert ausgeführt.

Durch die Verbindungszeit-Voraussageeinrichtung kann die Zeitgrenze für die Verarbeitung genauer vorgegeben werden, wenn die Verbindungszeit zwischen Computern nicht vernachlässigbar ist.





Die Computerauswahleinrichtung verhindert, daß eine Anzahl von Verarbeitungen mit kurzer Zeitgrenze auf einen einzelnen Computer konzentriert wird, wodurch es möglich ist, Echtzeiteigenschaften strenger einzuhalten.

Durch die Programmauswahleinrichtung werden mehrere Programme mit verschiedenen Dringlichkeitswerten bereitgestellt, damit eine Verarbeitungsanforderung an das Programm mit dem geeignetsten Dringlichkeitswert unter ihnen ausgegeben wird, selbst ohne Verwendung einer Prioritätswert-Änderungseinrichtung, mit dem Ergebnis, daß die Verarbeitung mit gewünschter Geschwindigkeit ausgeführt werden kann.

Ferner kann unter Verwendung der Verarbeitungsverlegeeinrichtung eine Verarbeitung, wie sie von einem aktuell betriebenen Programm gewünscht wird, auf ein anderes Programm verlegt und von diesem ausgeführt werden, das diese Verarbeitung mit im wesentlichen derselben Geschwindigkeit ausführen kann, ohne daß auf den Abschluß der aktuell ausgeführten Verarbeitung gewartet werden muß, wodurch eine strengere Einhaltung von Echtzeiteigenschaften gewährleistet ist.

Durch die Verarbeitungszeit-Voraussageeinrichtung und die Zeitgrenze-Einstelleinrichtung wird ermittelt, ob eine einer bestimmten Verarbeitung auferlegte Zeitgrenze erfüllt ist, wenn ein Programm diese Verarbeitung mit dem aktuellen Prioritätswert ausführen soll. Wenn erkannt wird, daß die Zeitgrenze nicht erfüllt wird, kann dem Programm eine kürzere Zeitgrenze zugewiesen werden. Unter Verwendung der neu eingestellten Zeitgrenze können die Prioritätswert-Umsetzungseinrichtungen die Prioritätswert-Änderungseinrichtung dem Programm einen höheren Prioritätswert verleihen und die Verarbeitung an ein anderes Programm verlegen, das denselben Vorgang ausführt und das der neuen Zeitgrenze genügen kann, um dieses Programm dazu aufzufordern, die Verarbeitung auszuführen. Auf diese Weise kann die Verarbeitung so verwaltet werden, daß der ursprünglich vorgegebene Zeitgrenze genügt wird.

Zusammengefaßt gesagt, können gemäß der Erfindung örtliche Prioritätswerte, wie sie an die Verarbeitung in jeweiligen Computern vergeben sind, in einem gesamten verteilten Computersystem gleichmäßig verwaltet werden. Zusätzlich können strengere Echtzeiteigenschaften in einem verteilten Computersystem gewährleistet werden, das Computer beinhaltet, die sich hinsichtlich des Funktionsvermögens, der Auslastung und des Typs voneinander unterscheiden.

#### Patentansprüche

1. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, gekennzeichnet durch:

— eine Prioritätswert-Bestimmungseinrichtung (144, 145; 146, 147) zum Bestimmen eines Prioritätswerts zum Ausführen eines Programms unter Berücksichtigung des Funktionsvermögens und/oder der Last jedes Computers wie auch eines Dringlichkeitswerts für die vom Programm auszuführende Verarbeitung.

2. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, gekennzeichnet durch:

— eine Prioritätswert-Bestimmungseinrichtung (572, 573; 146, 147) zum Bestimmen eines Prioritätswerts zum Ausführen eines Pro-

gramms unter Berücksichtigung des Funktionsvermögens und/oder der Last jedes Computers wie auch einer Zeitgrenze und auch einer Verarbeitungsmenge für die vom Programm auszuführende Verarbeitung.

3. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, dadurch gekennzeichnet, daß jeder Computer folgendes beinhaltet:

— eine Prioritätswert-Umsetzungseinrichtung (144, 145) zum Ausführen einer Umsetzung zwischen einem Dringlichkeitswert und einem Prioritätswert für die Verarbeitung abhängig von der Funktionsfähigkeit und der Auslastung dieses Computers; und

— eine Prioritätswert-Änderungseinrichtung (146, 147) zum Ändern des Prioritätswerts eines Programms, das die Verarbeitung ausführt, abhängig vom Prioritätswert, wie er von der Prioritätswert-Umsetzungseinrichtung angegeben wird.

4. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, dadurch gekennzeichnet, daß jeder Computer folgendes beinhaltet:

— eine Prioritätswert-Umsetzungseinrichtung (572, 573) zum Ausführen einer Umsetzung zwischen einem Satz aus einer Zeitgrenze und einer Verarbeitungsmenge für die Verarbeitung abhängig von der Funktionsfähigkeit und der Auslastung dieses Computers; und

— eine Prioritätswert-Änderungseinrichtung (146, 147) zum Ändern des Prioritätswerts eines Programms, das die Verarbeitung ausführt, abhängig vom Prioritätswert, wie er von der Prioritätswert-Umsetzungseinrichtung angegeben wird.

5. Verteiltes Computersystem nach Anspruch 3, dadurch gekennzeichnet, daß die Prioritätswert-Umsetzungseinrichtung (144, 145) eine Prioritätswert-Entsprechungstabelle (150, 151) beinhaltet, die die Entsprechung zwischen einem Dringlichkeitswert und einem Prioritätswert für die Verarbeitung angibt.

6. Verteiltes Computersystem nach Anspruch 3, dadurch gekennzeichnet, daß die Prioritätswert-Umsetzungseinrichtung (144, 145) die Entsprechungsbeziehung zwischen dem Dringlichkeitswert und dem Prioritätswert der Verarbeitung abhängig von Änderungen der Auslastungen der Computer verändert.

7. Verteiltes Computersystem nach Anspruch 6, dadurch gekennzeichnet, daß die Prioritätswert-Umsetzungseinrichtung (144, 145) den Prioritätswert jedes Programms auf den Computern abhängig von Änderungen der Auslastungen der Computer ändert.

8. Verteiltes Computersystem nach Anspruch 3, dadurch gekennzeichnet, daß jedes der mehreren Programme einen Dringlichkeitswert für die Verarbeitung spezifiziert, wenn dieses Programm andere Programme dazu auffordert, die Verarbeitung auszuführen.

9. Verteiltes Computersystem nach Anspruch 4, dadurch gekennzeichnet, daß jedes der mehreren Programme eine Zeitgrenze für die Verarbeitung spezifiziert, wenn dieses Programm ein anderes Programm dazu auffordert, die Verarbeitung auszuführen.

10. Verteiltes Computersystem nach Anspruch 4,



ferner gekennzeichnet durch:

- eine Verbindungszeit-Voraussageeinrichtung (670, 671) zum Voraussagen einer Verbindungszeit, wie sie für die Verbindung zu einem anderen Computer erforderlich ist; 5
- wobei jedes der mehreren Programme dann, wenn es ein anderes Programm zum Ausführen der Verarbeitung auffordert, von der Verbindungszeit-Voraussageeinrichtung eine vorausgesagte Verbindungszeit empfängt, wie sie dazu erforderlich ist, mit einem Computer Information auszutauschen, auf dem das andere Programm läuft, um die Zeitgrenze für die Verarbeitung unter Berücksichtigung der vorausgesagten Verbindungszeit zu spezifizieren. 15

11. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, gekennzeichnet durch:

- eine Computerauswahleinrichtung (294) zum Bestimmen eines Computers, der dazu geeignet ist, ein Programm auszuführen, und zwar unter Berücksichtigung des Funktionsvermögens und/oder der Auslastung jedes Computers und/oder eines Dringlichkeitswerts für die vom Programm auszuführende Verarbeitung; und 25
- eine Prioritätswert-Umsetzeinrichtung (260) zum Ausführen einer Umsetzung zwischen dem Dringlichkeitswert für die Verarbeitung und einem Prioritätswert für einen ausgewählten Computer, abhängig vom Funktionsvermögen und/oder der Auslastung des Computers. 30

12. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, gekennzeichnet durch:

- eine Computerauswahleinrichtung (294) zum Bestimmen eines Computers, der dazu geeignet ist, ein Programm auszuführen, und zwar unter Berücksichtigung des Funktionsvermögens und/oder der Auslastung jedes Computers und/oder eines Satzes aus einer Zeitgrenze und einer Verarbeitungsmenge für die vom Programm auszuführende Verarbeitung; und 45
- eine Prioritätswert-Umsetzeinrichtung (260) zum Ausführen einer Umsetzung zwischen dem Satz aus einer Zeitgrenze und einer Verarbeitungsmenge für die Verarbeitung und einem Prioritätswert für einen ausgewählten Computer, abhängig vom Funktionsvermögen und/oder der Auslastung des Computers. 50

13. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, wobei ein aktuell auf einem Computer laufendes Programm auf einen anderen Computer verschoben wird, gekennzeichnet durch:

- eine Computerauswahleinrichtung (294) zum Bestimmen eines Zielcomputers, auf den das Programm zu verschieben ist, und zwar unter Berücksichtigung des Funktionsvermögens und/oder der Auslastung jedes Computers und/oder des Dringlichkeitswerts für die vom Programm auszuführende Verarbeitung; 65
- und
- eine Prioritätswert-Umsetzeinrichtung (260) zum Ausführen einer Umsetzung zwi-

schen dem Dringlichkeitswert der Verarbeitung und einem Prioritätswert auf dem Zielcomputer abhängig vom Funktionsvermögen und der Auslastung des Zielcomputers.

14. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (130—136) laufen, gekennzeichnet durch:

- eine Prioritätswert-Umsetzeinrichtung (260) zum Ausführen einer Umsetzung zwischen einem Dringlichkeitswert für die von einem Programm ausgeführte Verarbeitung und einem Prioritätswert auf jedem Computer, abhängig vom Funktionsvermögen und der Auslastung jedes Computers; und
- eine Computerauswahleinrichtung (294) zum Auswählen eines Computers, der zum Ausführen des Programms geeignet ist, unter Berücksichtigung der Auslastung jedes Computers und des Prioritätswerts für jeden Computer, wie durch die Prioritätswert-Umsetzeinrichtung angegeben. 20

15. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (430—442) laufen, wobei mehrere denselben Vorgang ausführende Programme in den Computern vorhanden sind und wobei eines der mehreren denselben Vorgang ausführenden Programme dazu aufgefordert wird, eine Verarbeitung auszuführen, gekennzeichnet durch:

- eine Programmauswahleinrichtung (410) zum Bestimmen des Programms mit optimalem Prioritätswert aus den mehreren denselben Vorgang ausführenden Programmen unter Berücksichtigung des Funktionsvermögens und/oder der Auslastung jedes Computers und/oder des Dringlichkeitswerts der Verarbeitung. 30

16. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (430—442) laufen, wobei mehrere denselben Vorgang ausführende Programme in den Computern vorhanden sind und wobei eines der mehreren denselben Vorgang ausführenden Programme dazu aufgefordert wird, eine Verarbeitung auszuführen, gekennzeichnet durch:

- eine Programmauswahleinrichtung (410) zum Bestimmen des Programms mit optimalem Prioritätswert aus den mehreren denselben Vorgang ausführenden Programmen unter Berücksichtigung des Funktionsvermögens und/oder der Auslastung jedes Computers und/oder eines Satzes aus einer Zeitgrenze und einer Verarbeitungsmenge der Verarbeitung. 45

17. Verteiltes Computersystem mit mehreren Computern (100—103), auf denen mehrere Programme (430—442) laufen, wobei mehrere denselben Vorgang ausführende Programme in den Computern vorhanden sind und wobei eines der mehreren denselben Vorgang ausführenden Programme, das gerade eine Verarbeitung ausführt, zusätzliche aufgefordert wird, eine andere Verarbeitung auszuführen, gekennzeichnet durch:

- eine Programmauswahleinrichtung (410) zum Bestimmen des Programms mit optimalem Prioritätswert aus den mehreren denselben Vorgang ausführenden Programmen unter Berücksichtigung des Funktionsvermögens 50



und/oder der Auslastung jedes Computers und/oder des Dringlichkeitswerts der Verarbeitung;

– eine Verarbeitungsverlegeeinrichtung (550) zum Verlegen der zusätzlich angeforderten Verarbeitung auf das Programm, wie es von der Programmauswahleinrichtung angezeigt wird.

18. Verteiltes Computersystem mit mehreren Computern (100–103), auf denen mehrere Programme (430–442) laufen, wobei mehrere denselben Vorgang ausführende Programme in den Computern vorhanden sind und wobei eines der mehreren denselben Vorgang ausführenden Programme, das gerade eine Verarbeitung ausführt, zusätzliche aufgefördert wird, eine andere Verarbeitung auszuführen, gekennzeichnet durch:

– eine Programmauswahleinrichtung (760) zum Bestimmen des Programms mit optimalem Prioritätswert aus den mehreren denselben Vorgang ausführenden Programmen unter Berücksichtigung des Funktionsvermögens und/oder der Auslastung jedes Computers und/oder eines Satzes aus einer Zeitgrenze und einer Verarbeitungsmenge der Verarbeitung;

– eine Verarbeitungsverlegeeinrichtung (550) zum Verlegen der zusätzlich angeforderten Verarbeitung auf das Programm, wie es von der Programmauswahleinrichtung angezeigt wird.

19. Verteiltes Computersystem mit mehreren Computern (100–103), auf denen mehrere Programme (430–442) laufen, wobei mehrere denselben Vorgang ausführende Programme in den Computern vorhanden sind, gekennzeichnet durch:

– eine Verarbeitungs-Voraussageeinrichtung (800) zum Voraussagen der Verarbeitungszeit, wie sie wahrscheinlich ein Programm benötigt, um eine gerade bearbeitete Verarbeitung zu beenden, und zwar auf Grundlage des Funktionsvermögens, der Auslastung und der Verarbeitungsmenge für einen Computer, in dem das Programm vorliegt;

– eine Zeitgrenze-Einstelleinrichtung (802) zum Berechnen einer neuen Zeitgrenze, wenn eine vorausgesagte Verarbeitungszeit, wie sie durch die Verarbeitungszeit-Voraussageeinrichtung angegeben wird, die Zeitgrenze nicht erfüllt, wie sie für diese Verarbeitung erforderlich ist, und zwar ausgehend von der Zeit, wie sie verstrichen ist, seit die Verarbeitung angefordert wurde, und der erforderlichen Zeitgrenze;

– eine Prioritätswert-Umsetzeinrichtung (580, 581) zum Ausführen einer Umsetzung zwischen einem Satz aus der neuen Zeitgrenze und der Verarbeitungsmenge für das Programm und dem Prioritätswert in jedem Computer;

– eine Programmauswahleinrichtung (760) zum Bestimmen eines Programms, das den mehreren denselben Vorgang ausführenden Programmen, das denselben oder einen höheren Prioritätswert aufweist, als es dem Prioritätswert entspricht, wie er von der Prioritätswert-Umsetzeinrichtung angegeben wird; und

– eine Verarbeitungsverlegeeinrichtung (550)

zum Verlegen der Verarbeitungsanforderung an das durch die Programmauswahleinrichtung angegebene Programm.

20. Verteiltes Computersystem mit mehreren Computern (100–103), auf denen mehrere Programme (430–442) laufen, gekennzeichnet durch:

– eine Verarbeitungszeit-Voraussageeinrichtung (800) zum Voraussagen der Verarbeitungszeit, wie sie vermutlich dazu erforderlich ist, eine von den mehreren Programmen gerade ausgeführte Verarbeitung abzuschließen, und zwar auf Grundlage des Funktionsvermögens, der Auslastung und der Verarbeitungsmenge des jeweiligen Computers, in dem ein jeweiliges Programm vorhanden ist;

– eine Zeitgrenze-Einstelleinrichtung (802) zum Berechnen einer neuen Zeitgrenze, wenn eine vorausgesagte Verarbeitungszeit, wie sie durch die Verarbeitungszeit-Voraussageeinrichtung angegeben wird, die Zeitgrenze nicht erfüllt, wie sie für diese Verarbeitung erforderlich ist, und zwar ausgehend von der Zeit, wie sie verstrichen ist, seit die Verarbeitung angefordert wurde, und der erforderlichen Zeitgrenze;

– eine Prioritätswert-Umsetzeinrichtung (580, 581) zum Ausführen einer Umsetzung zwischen einem Satz aus der neuen Zeitgrenze und der Verarbeitungsmenge für das Programm und dem Prioritätswert in jedem Computer; und

– eine Prioritätswert-Änderungseinrichtung (146, 147) zum Ändern des Prioritätswerts eines Programms, das die Verarbeitung ausführt.

Hierzu 35 Seite(n) Zeichnungen



- Leerseite -

FIG.. 1

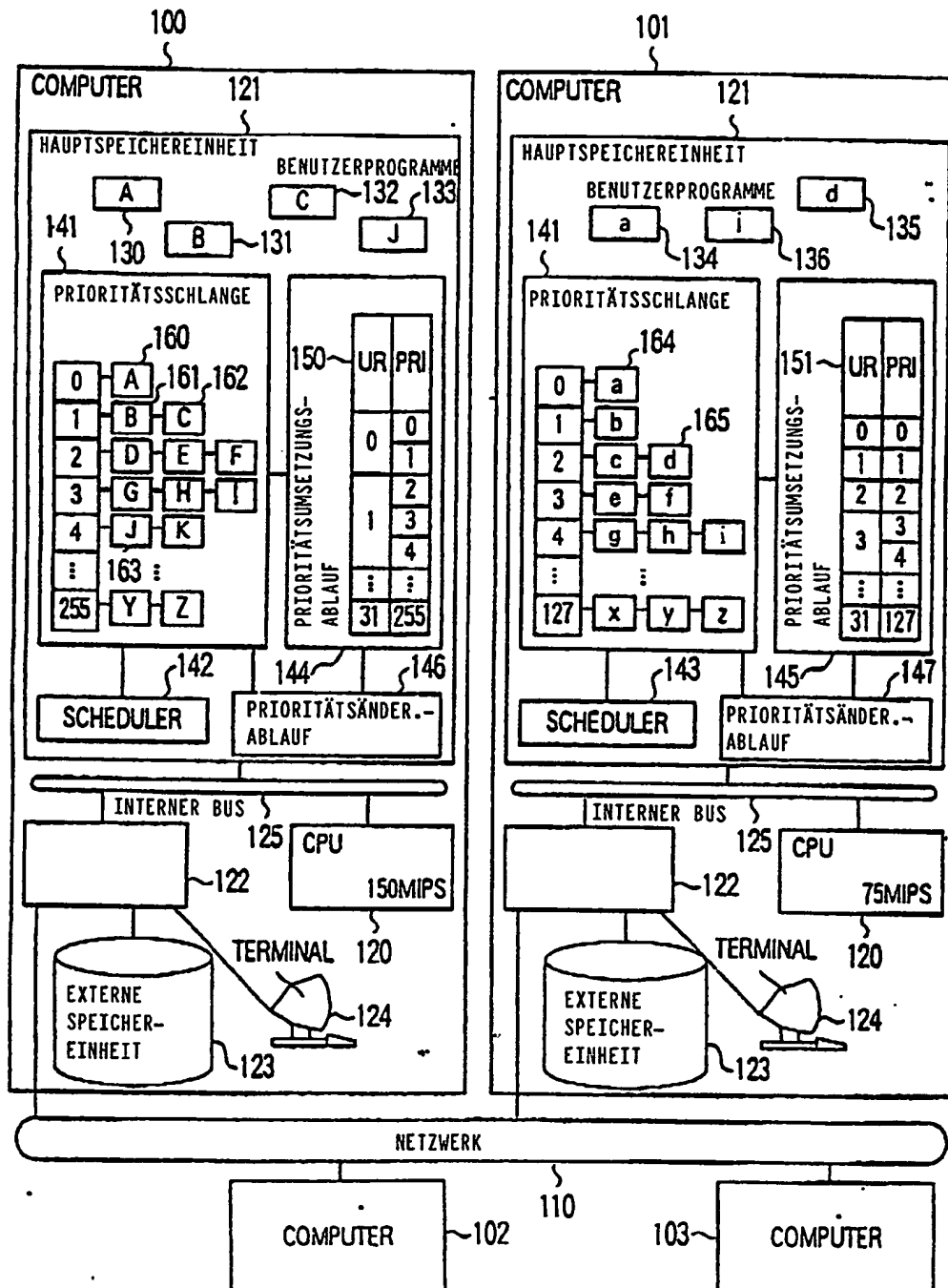


FIG. 2

```
int to_urgency[256]: /* priority -> urgency */
```

170

```
struct {
    int start: /* start of range */
    int end: /* end of range */
} to_priority[32]: /* urgency -> priority */
```

171

172

PRIORITÄTSWERT-ENTSPRECHUNGSTABELLE			
DRINGLICHKEITS- WERT			PRIORITÄTSWERT
to_urgency[0]=0	0	0	to_priority[0].start=0
to_urgency[1]=0		1	to_priority[0].end=1
to_urgency[2]=1	1	2	to_priority[1].start=2
to_urgency[3]=1		3	
to_urgency[4]=1		4	to_priority[1].end=4
to_urgency[5]=2	2	5	to_priority[2].start=5
to_urgency[6]=2		6	
to_urgency[7]=2		7	to_priority[2].end=7
	:	:	
to_urgency[255]=0	31	255	to_priority[31].end=255

FIG. 3

DRINGLICHKEITS-  
WERT → PRIORITÄTSWERT

```
priority = SPEED/LOAD * urgency*urgency : ~180  
if (priority > 255) priority = 255 :
```

PRIORITÄTSWERT - DRINGLICHKEITSWERT

```
urgency = sqrt(priority * LOAD/SPEED) : ~181  
if (urgency > 31) urgency = 31 :
```

HINWEIS: Sqrt(x) REPRÄSENTIERT DAS QUADRAT VON X,  
LOAD REPRÄSENTIERT DIE AUSLASTUNG UND  
SPEED REPRÄSENTIERT DAS FUNKTIONSVERMÖGEN

FIG. 4

```
func (urgency, arg1, arg2, ...) ; ~190
```

```
func (arg1, arg2, ..., urgency) : ~191
```

FIG. 5

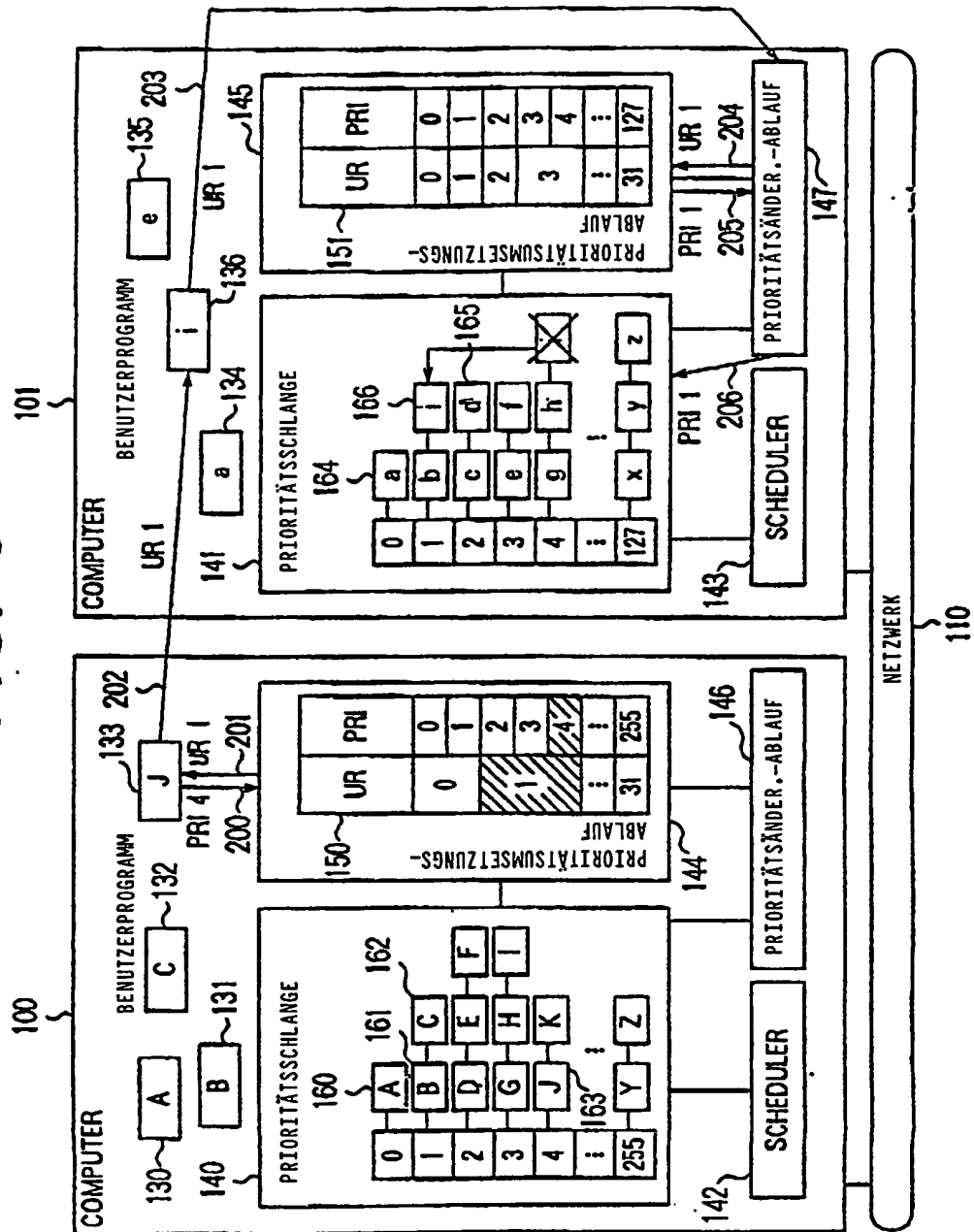




FIG. 6

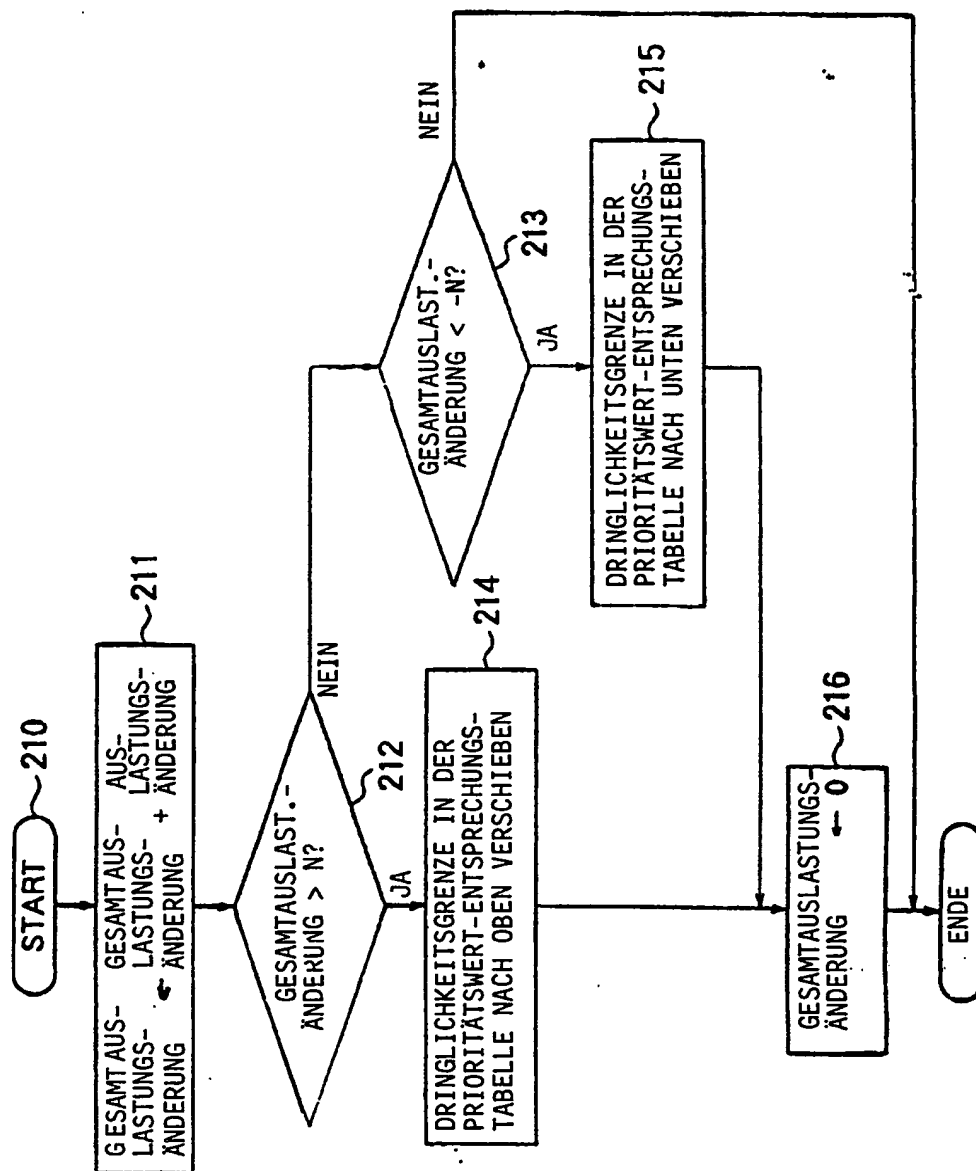
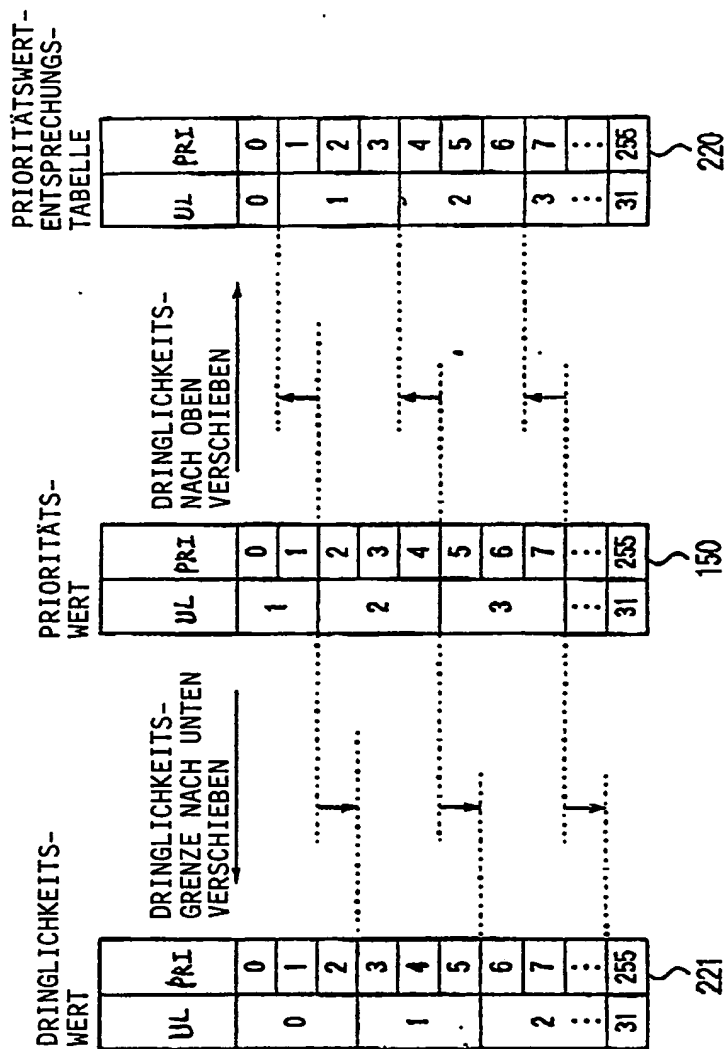


FIG. 7



## FIG. 8

```
for (u = 0 : u < 31 : u++) {  
  to_priority[u].end -- : ~ 240  
  to_priority[u+1].start -- : ~ 241  
  to_urgency[to_priority[u+1].start] ++ : ~ 242  
  change_all_priority(to_priority[u+1].start, -1) : ~ 243  
}
```

~ 230

```
for (u = 0 : u < 31 : u++) {  
  to_priority[u].end ++ : ~ 244  
  to_priority[u+1].start ++ : ~ 245  
  to_urgency[to_priority[u].end] -- : ~ 246  
  change_all_priority(to_priority[u].end, 1) : ~ 247  
}
```

~ 231

FIG. 9

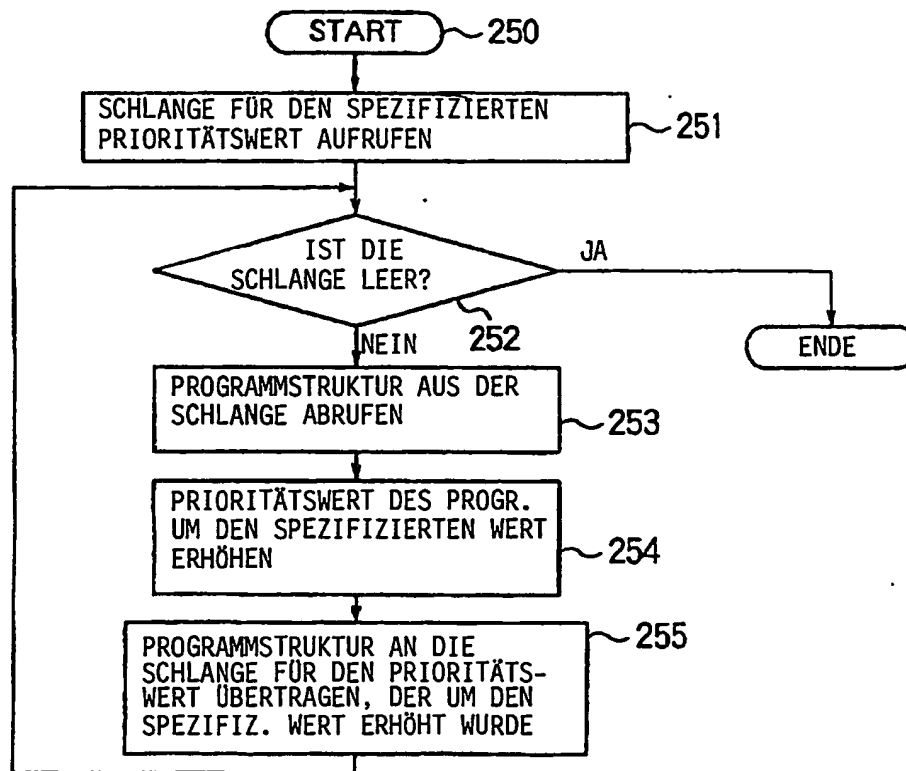


FIG. 10

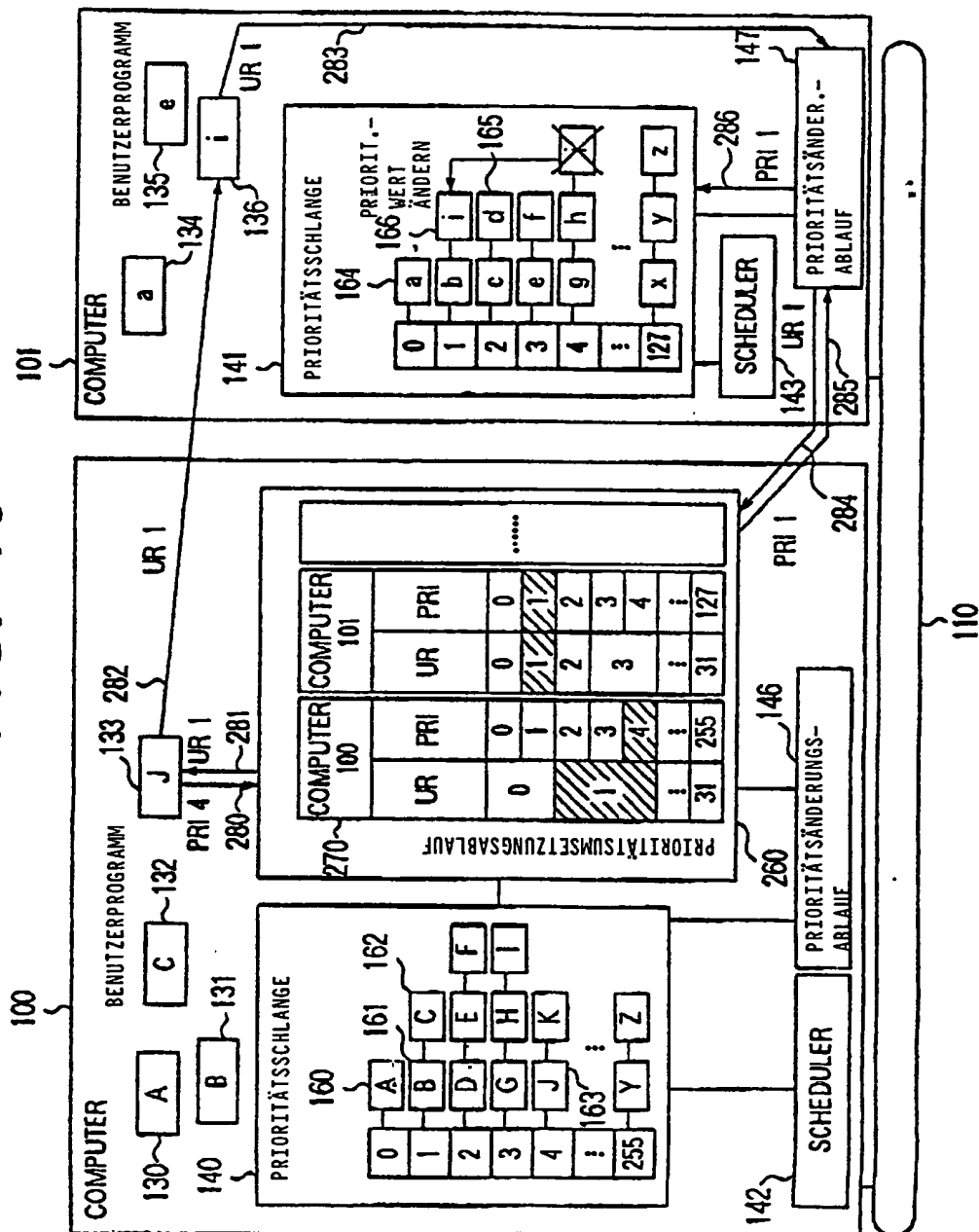


FIG. 11

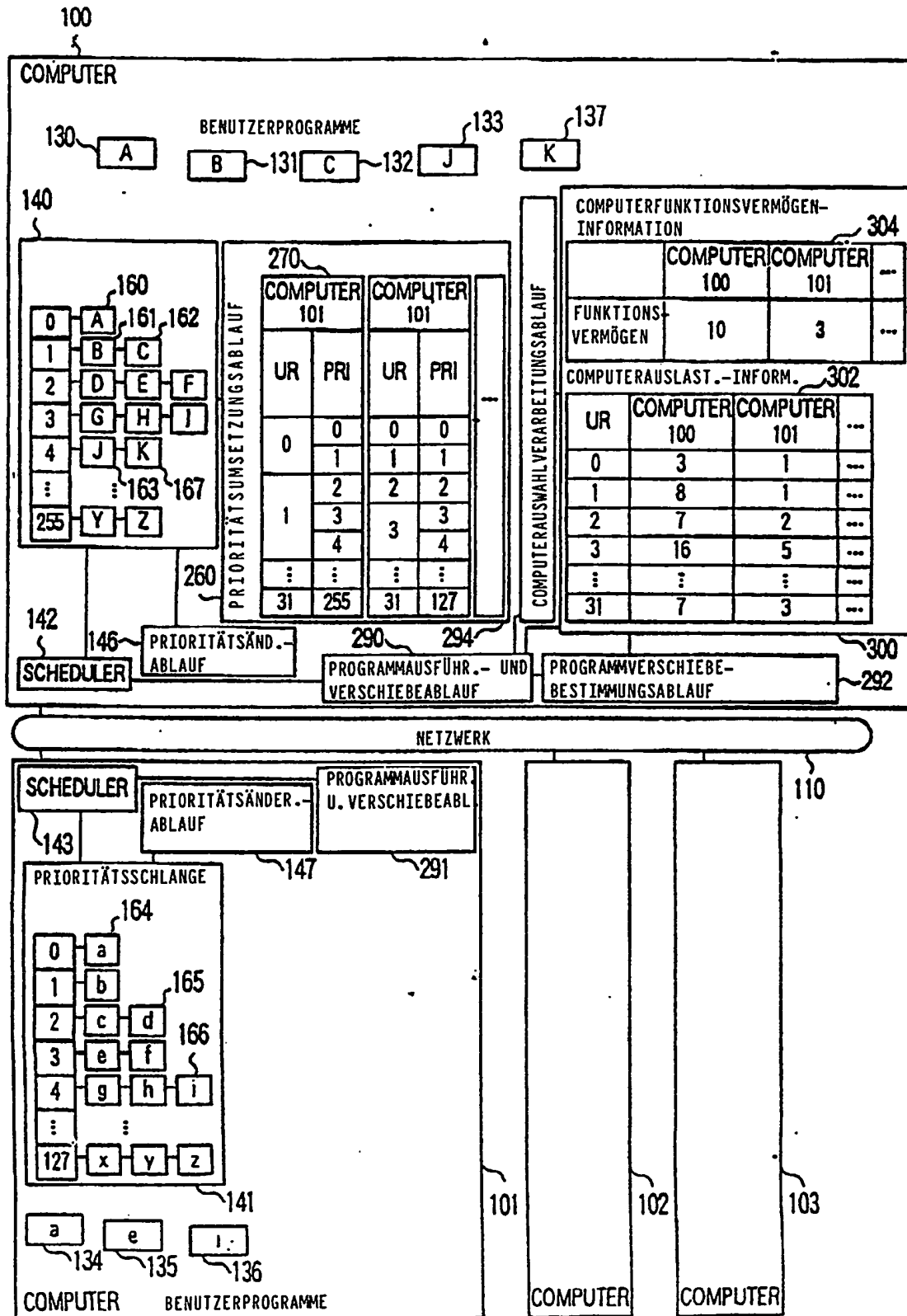


FIG. 12

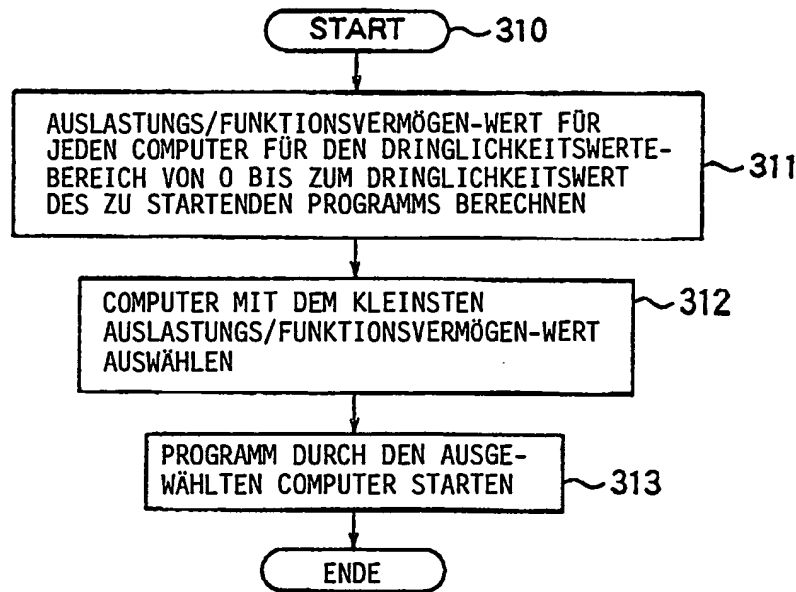




FIG. 13

302 COMPUTERAUSLASTUNG- INFORMATIONSTABELLE				306 LASTWERTTABELLE				308 AUSLASTUNGS/FUNKTIONSVERMÖGEN- WERT-TABELLE			
DRINGLICH- KEITSWERT	COMPUTER 100	COMPUTER 101	...	DRINGLICH- KEITSWERTE- BEREICH	COMPUTER 100	COMPUTER 101	...	DRINGLICH- KEITSWERTE- BEREICH	COMPUTER 100	COMPUTER 101	...
0	3	1	...	0	3	1	...	0	0.30	0.33	...
1	8	1	...	0~1	11	2	...	0~1	1.10	0.67	...
2	7	2	...	0~2	18	4	...	0~2	1.80	1.33	...
3	16	5	...	0~3	34	9	...	0~3	3.40	3.00	...
:	:	:	...	:	:	:	...	:	:	:	...
31	7	3	...	0~31	241	71	...	0~31	24.10	23.67	...



FIG. 14

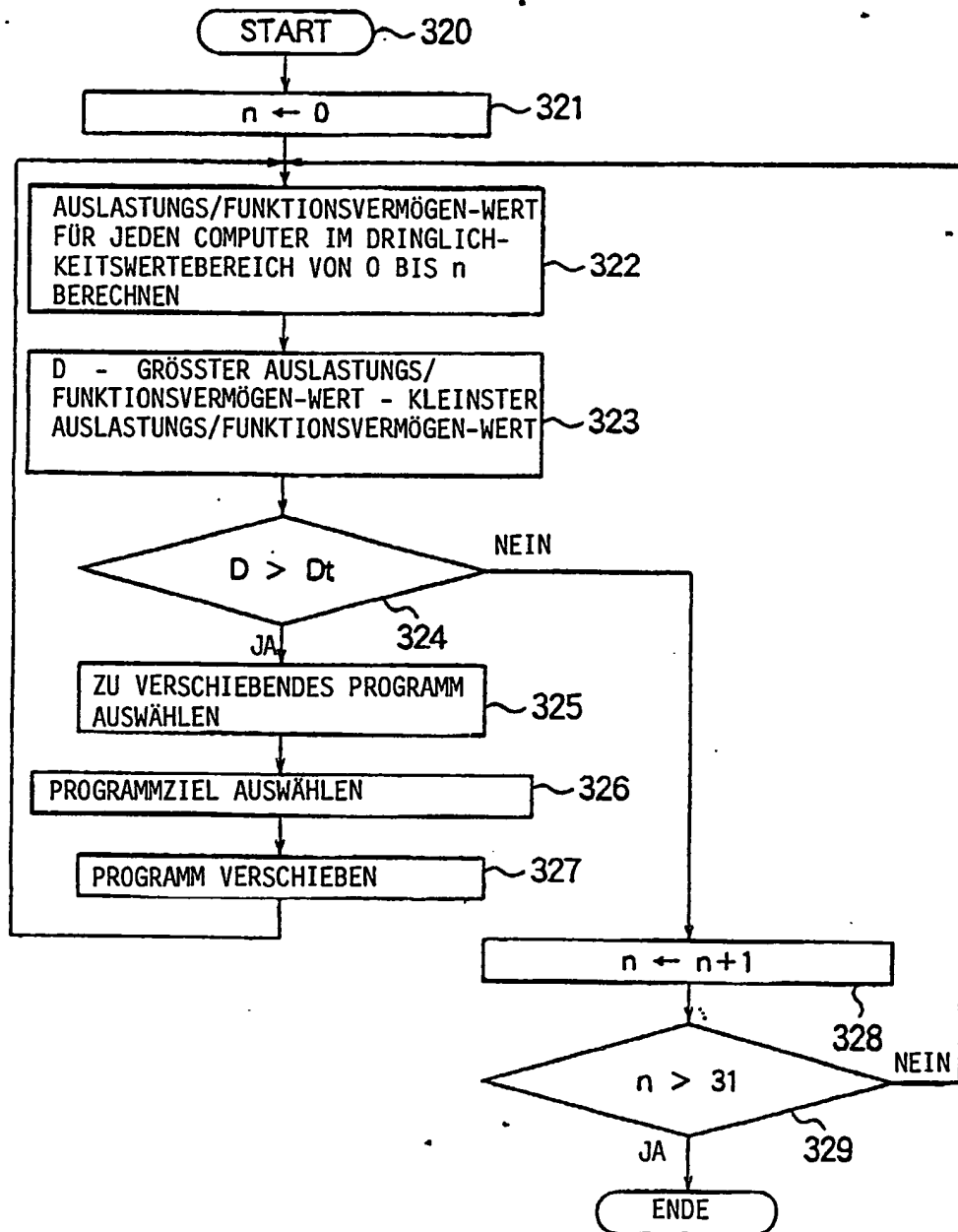


FIG. 15

COMPUTERAUSLASTUNGS- INFORMATIONSTABELLE				LASTWERTTABELLE				AUSLASTUNGS/FUNKTIONS- VERMÖGEN-WERT-TABELLE			
302				306				308			
DRINGLICH- KEITSWERT	COMPUTER 100	COMPUTER 101	...	DRINGLICH- KEITSWERT- BEREICH	COMPUTER 100	COMPUTER 101	...	DRINGLICH- KEITSWERT- BEREICH	COMPUTER 100	COMPUTER 101	...
0	3	1	...	0	3	1	...	0	0.30	0.33	...
1	7	2	...	0~1	10	3	...	0~1	1.00	1.00	...
2	7	2	...	0~2	17	5	...	0~2	1.70	1.67	...
3	16	5	...	0~3	33	10	...	0~3	3.30	3.33	...
:	:	:	...	:	:	:	...	:	:	:	...
31	7	3	...	0~31	240	72	...	0~31	24.00	24.00	...



FIG. 16

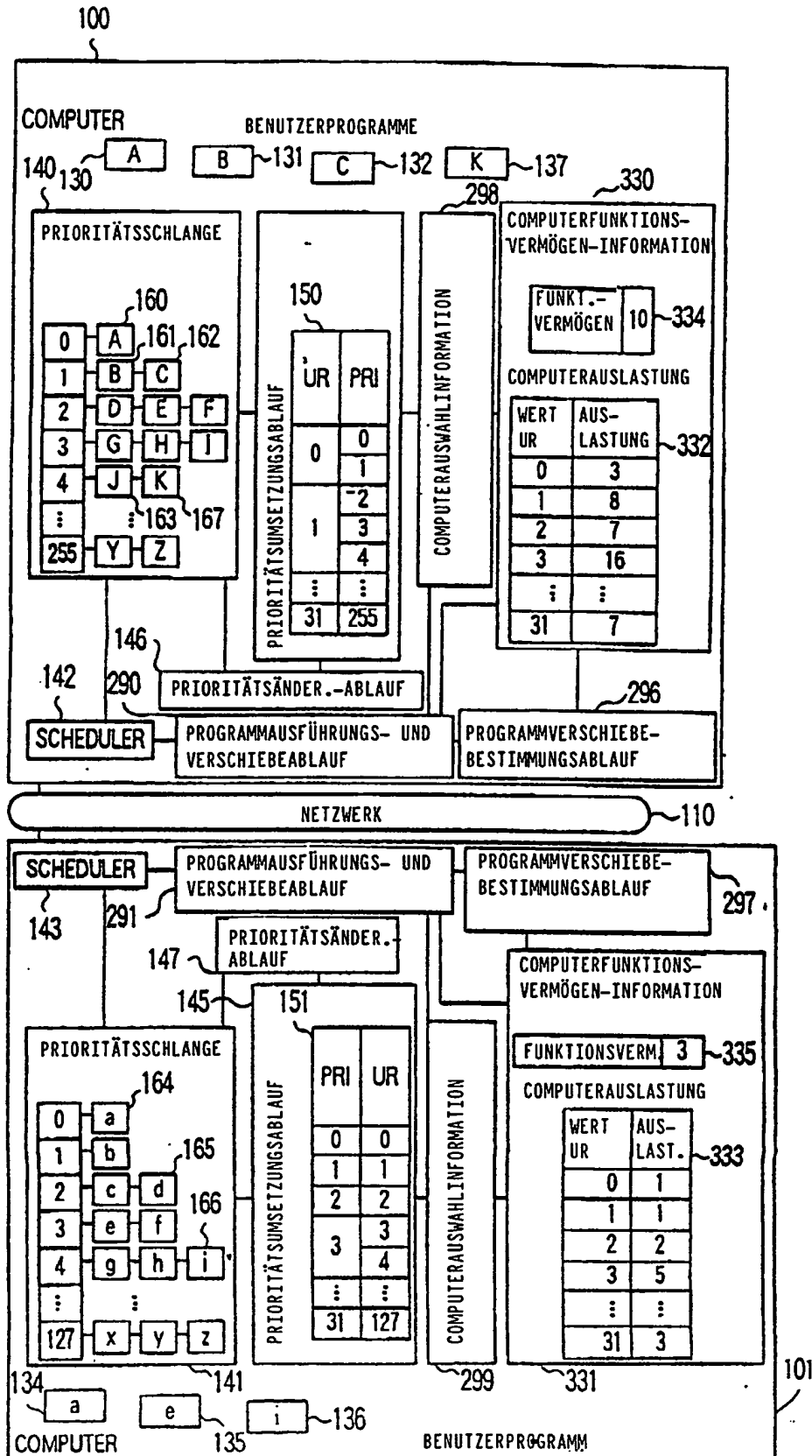


FIG. 17

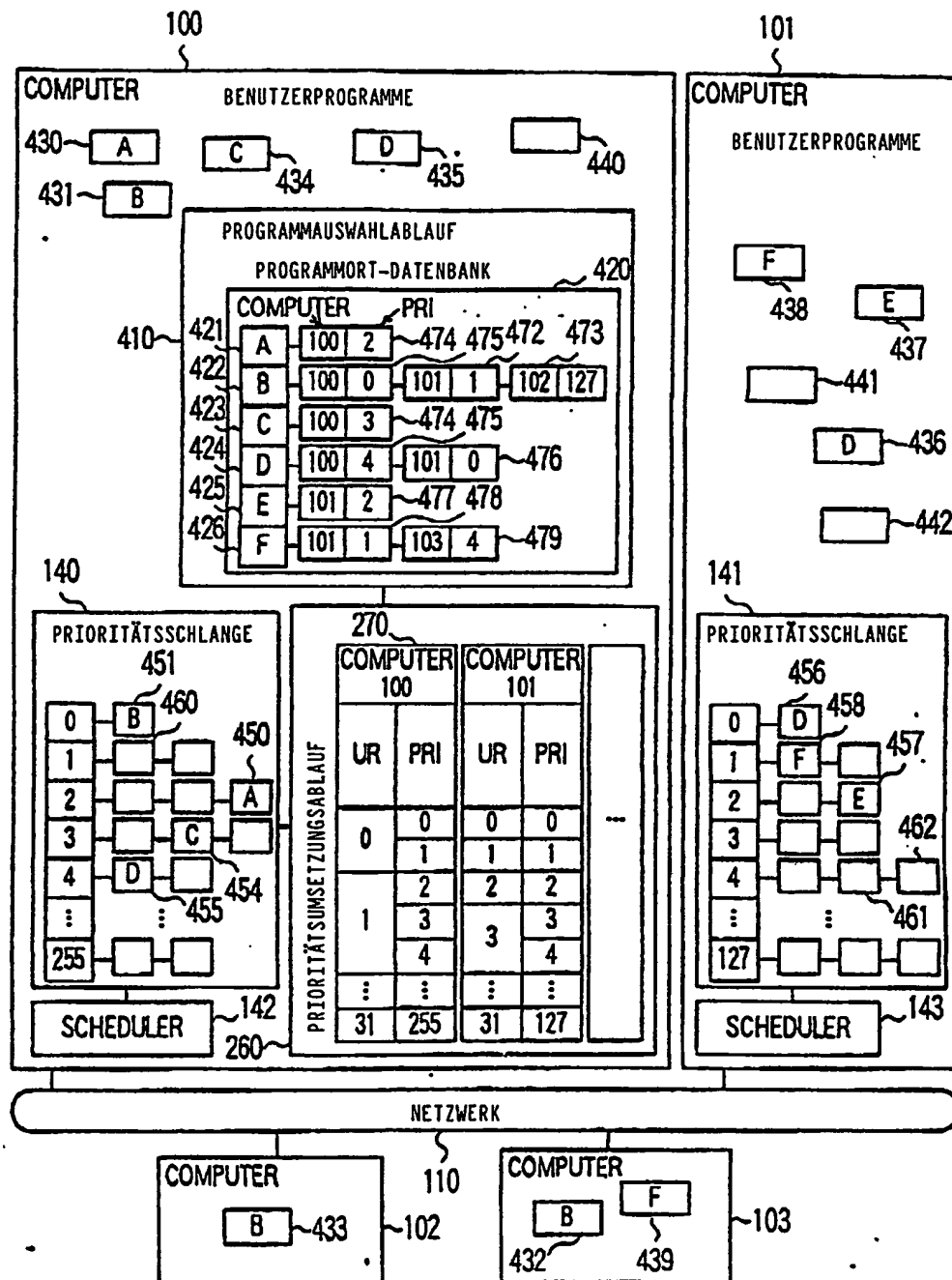


FIG. 18

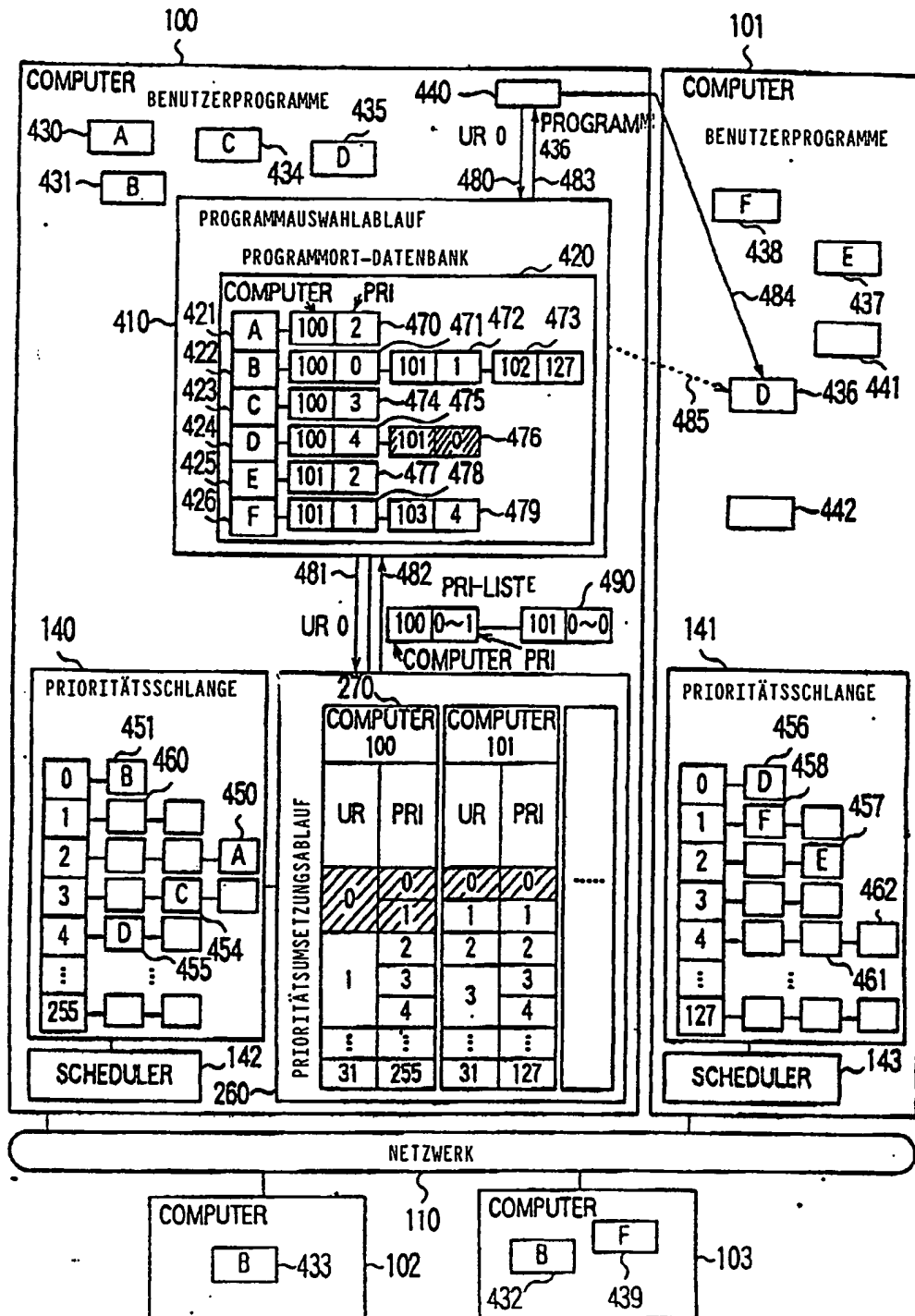


FIG. 19

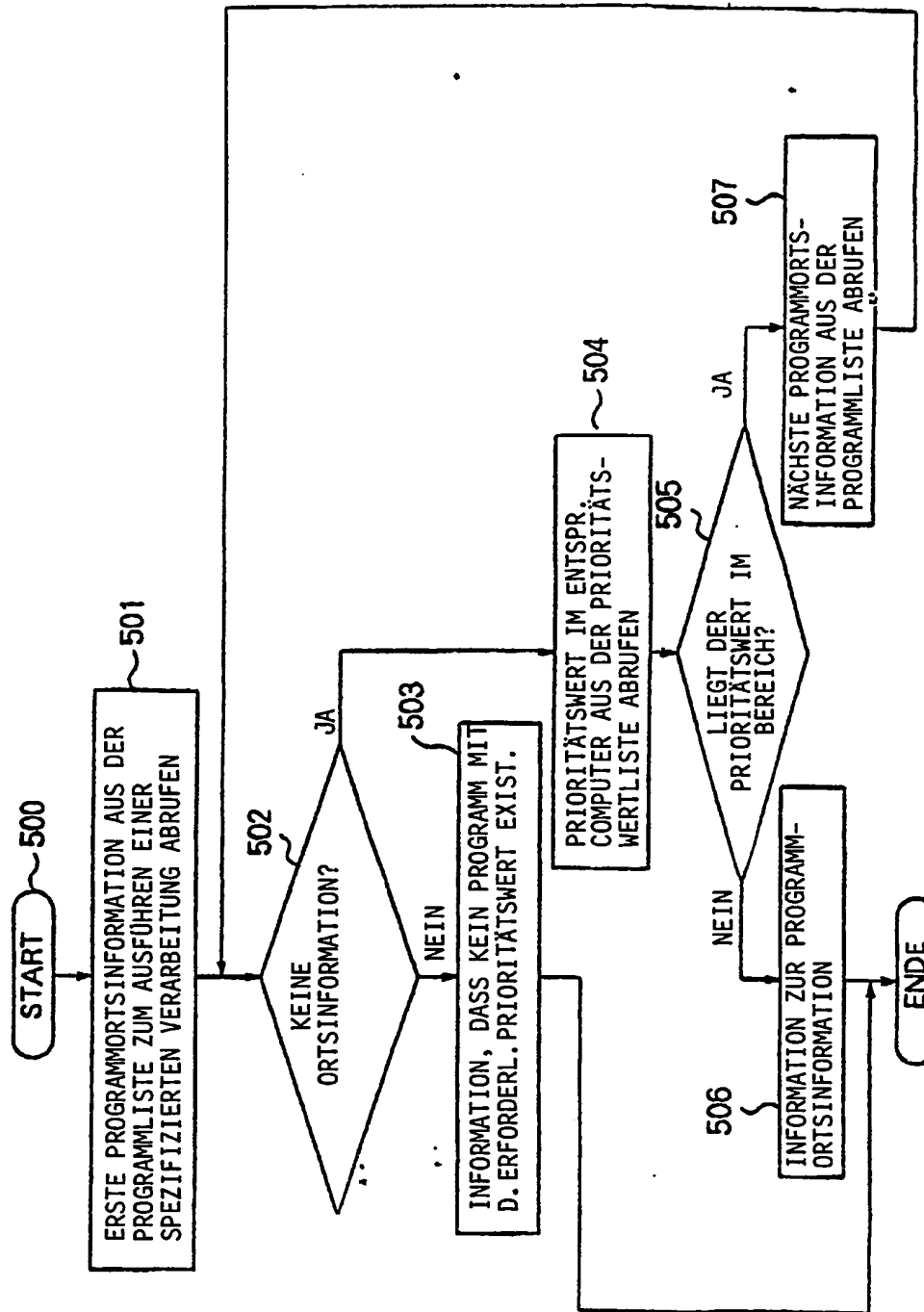




FIG. 20

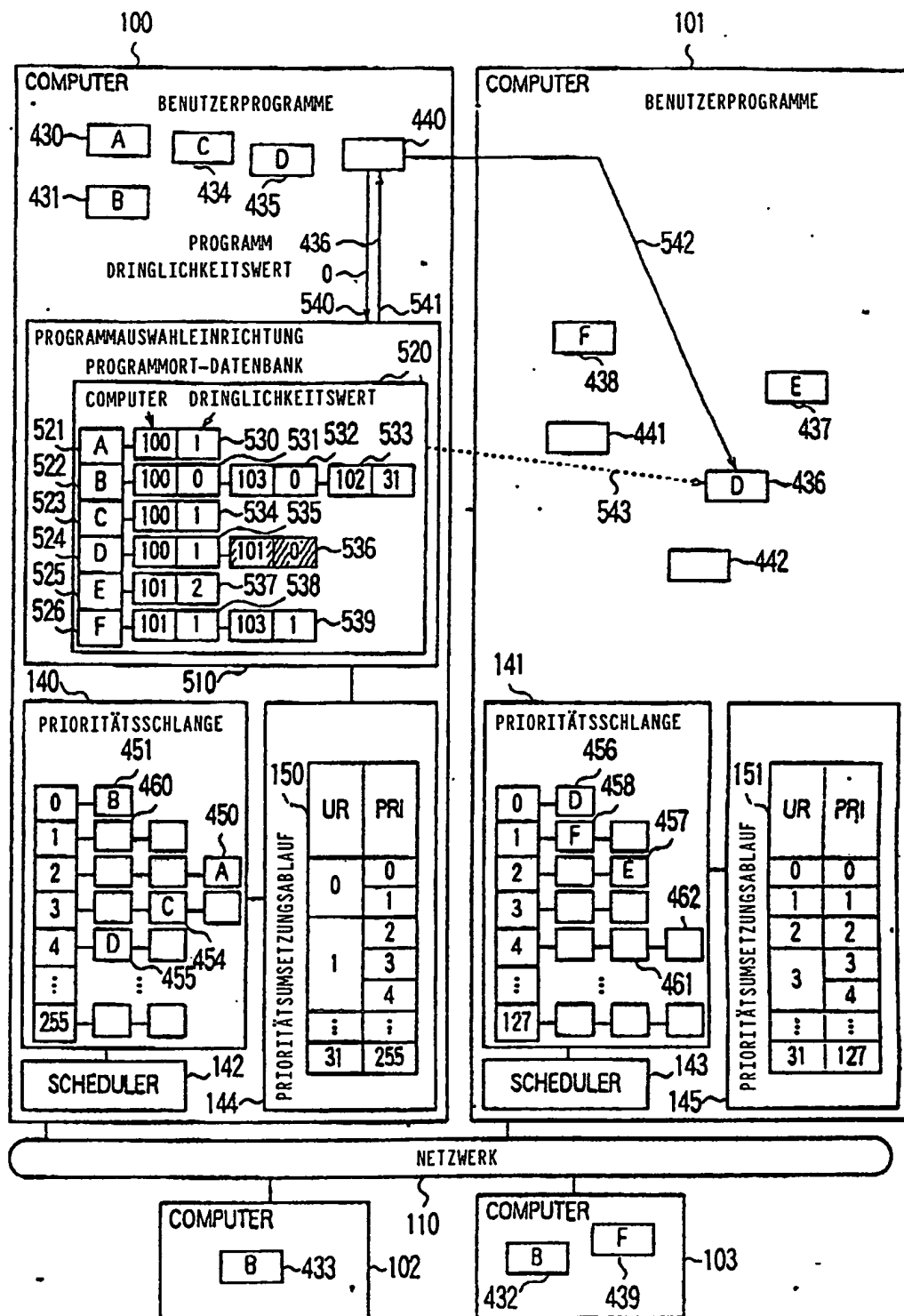


FIG. 21

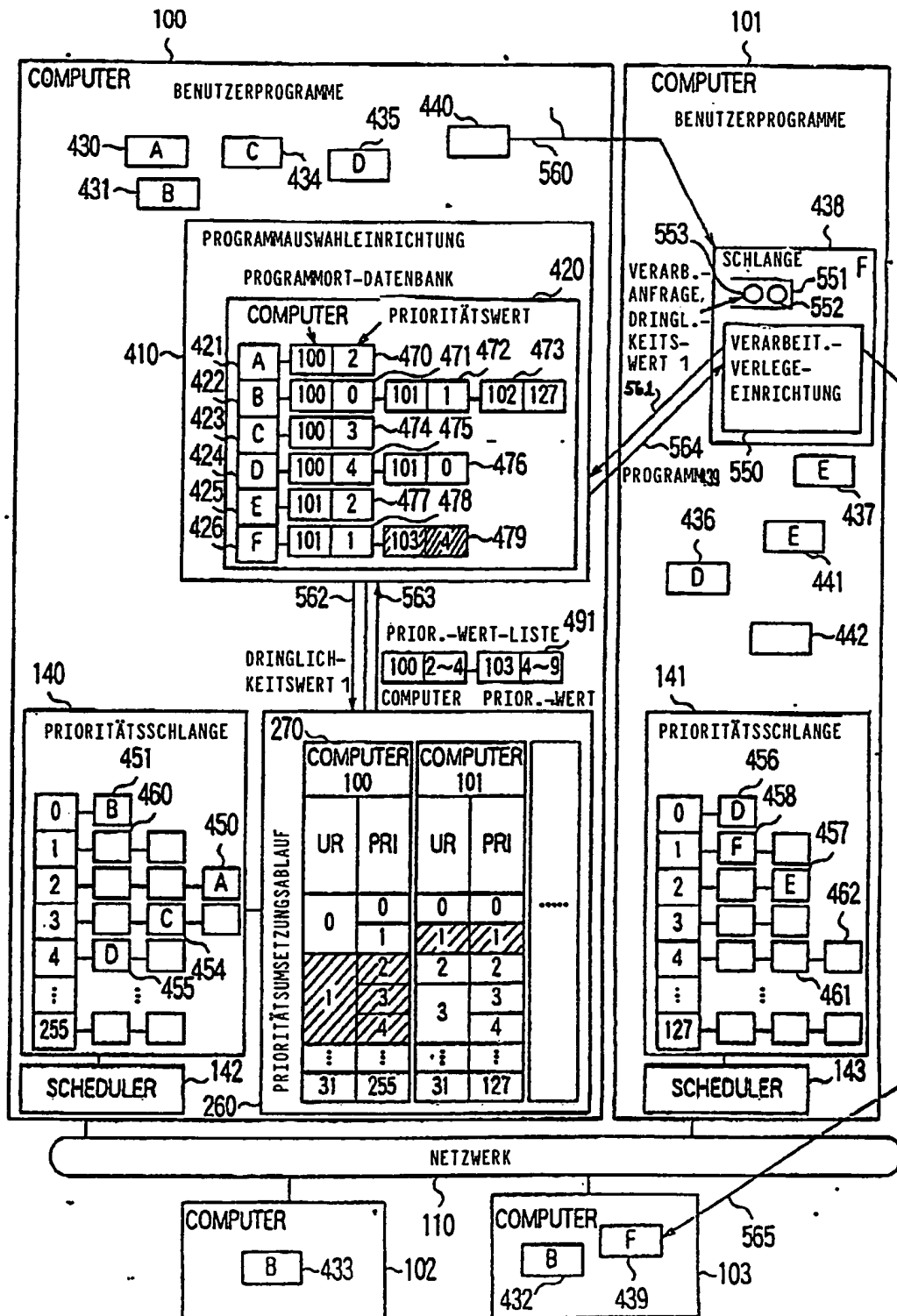


FIG. 22

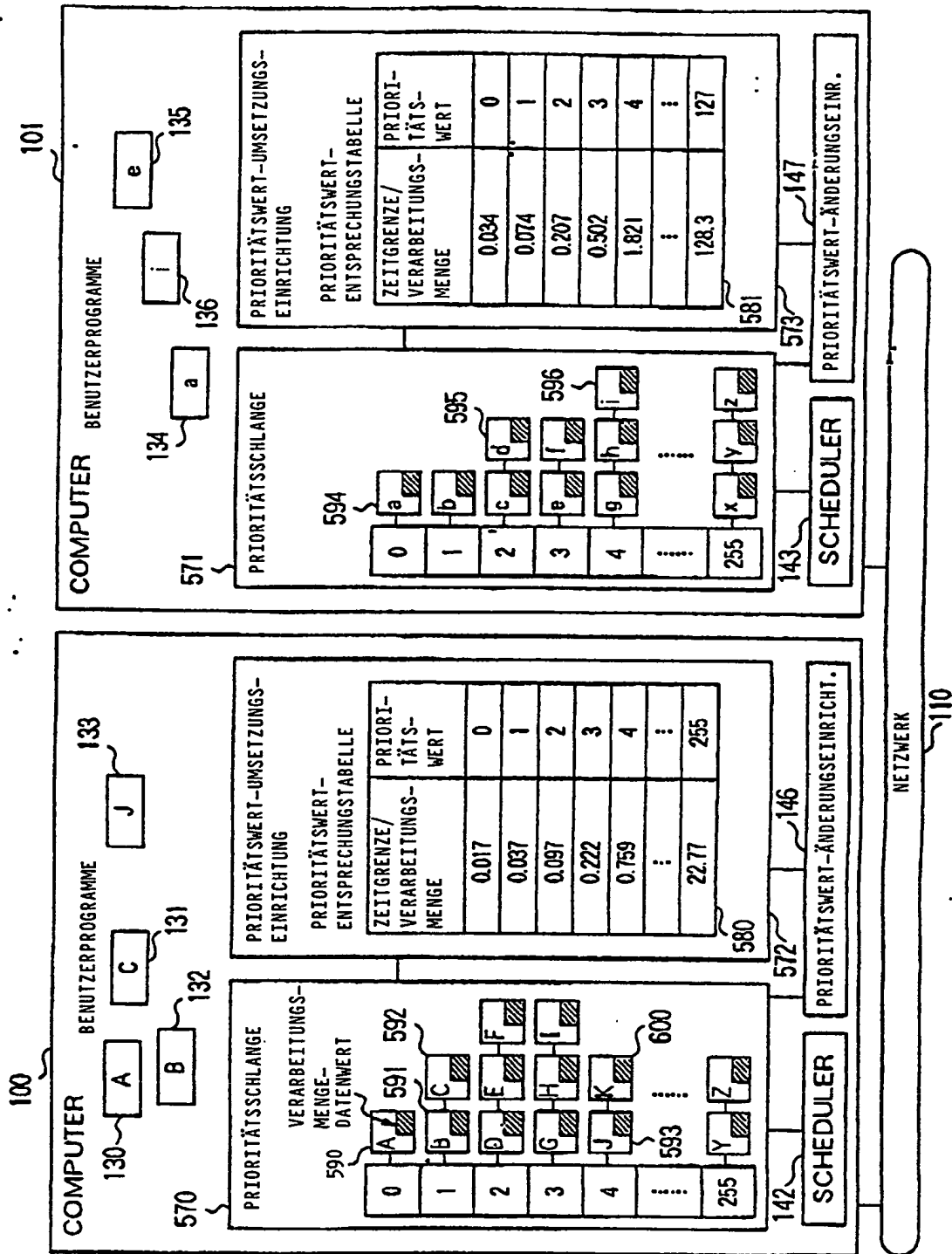


FIG. 23

```
double to_time[256]; /* priority -> time limit */
```

 610

FIG. 24

```
to_priority(timelimit)
double timelimit;
{
    int i = 0;
    int j = 255;
    int middle;
    while (i < j) {
        middle = (i + j) / 2;
        if (timelimit < to_time[middle])
            j = middle;
        else if (timelimit >= to_time[middle+1])
            i = middle + 1;
        else
            return middle;
    }
}
```

 620

FIG. 25

```
func(timelimit, arg1, arg2, ...);
```

 630

```
func(arg1, arg2, ..., timelimit);
```

 631

FIG. 26

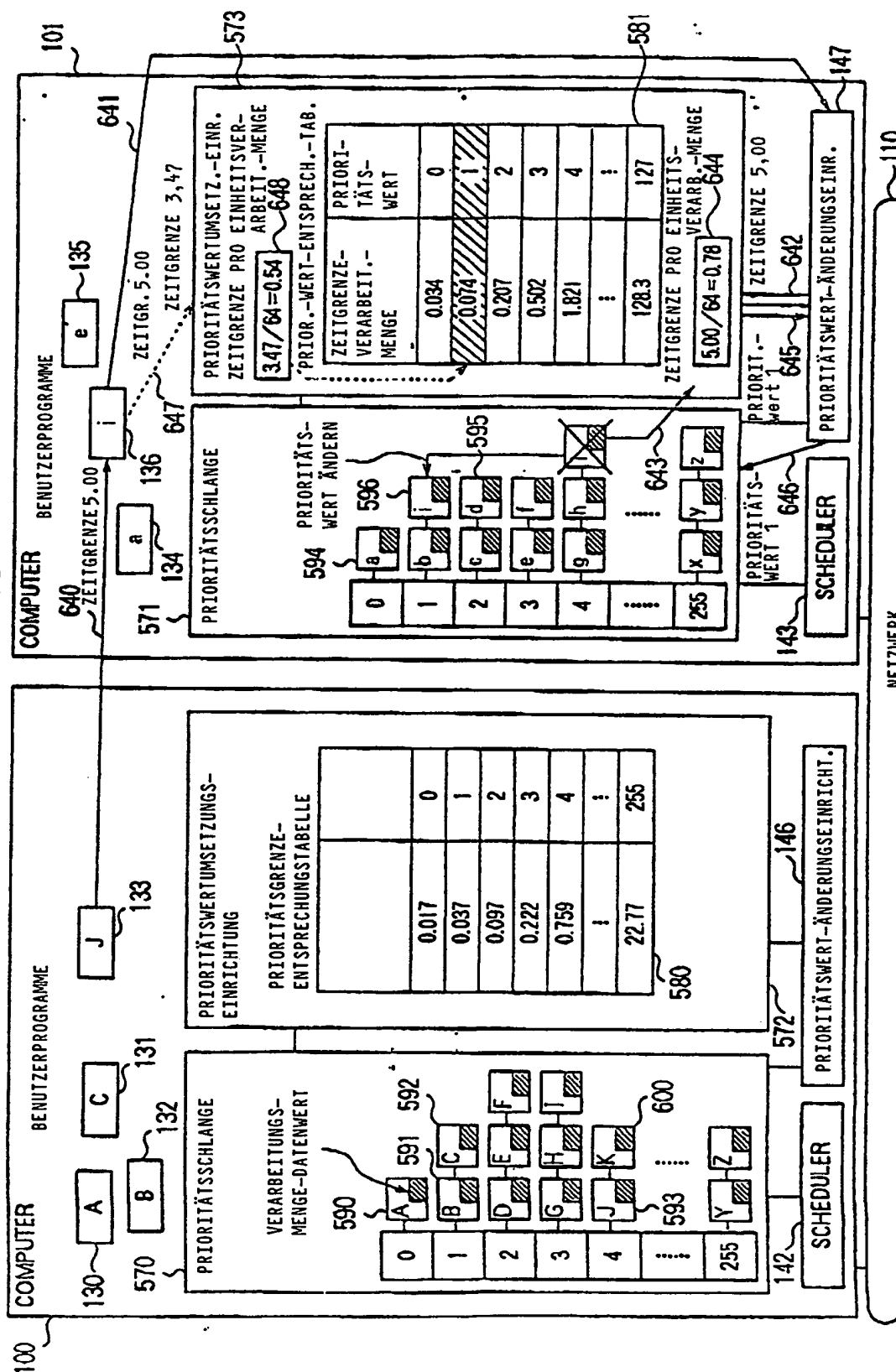


FIG. 27

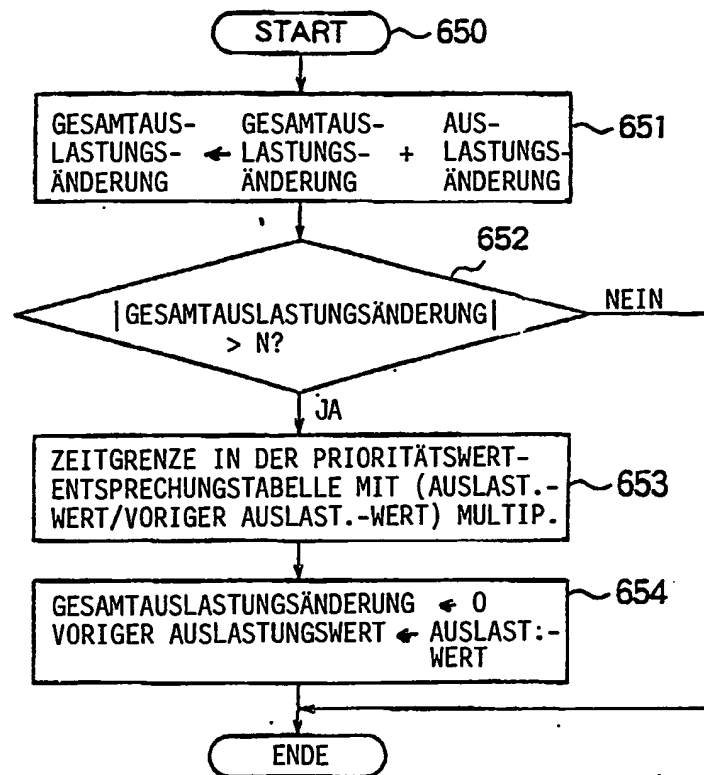


FIG. 28

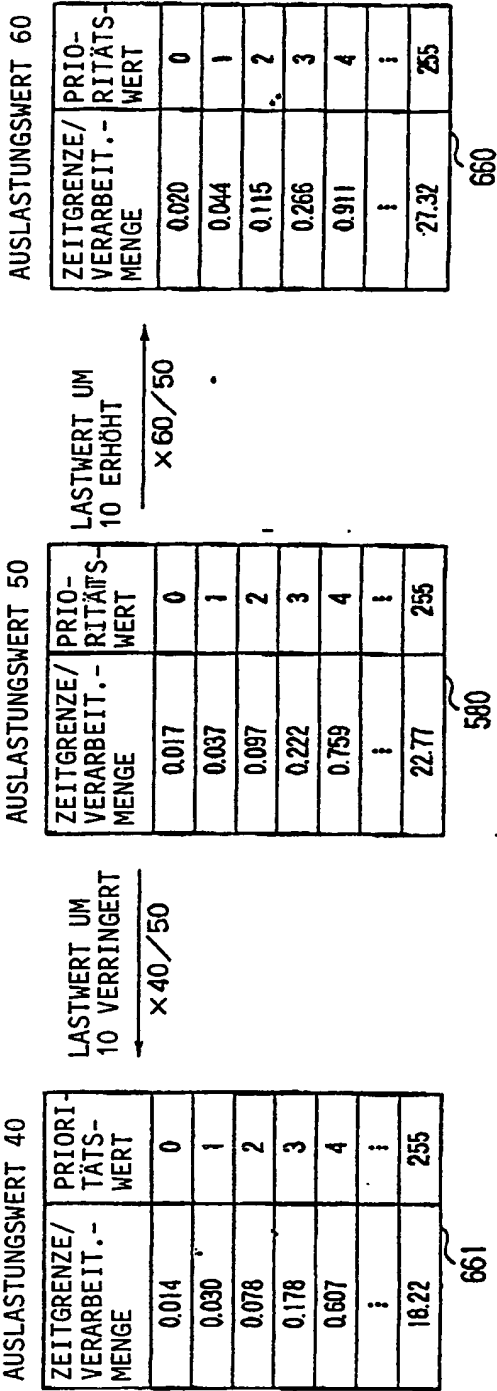




FIG. 29

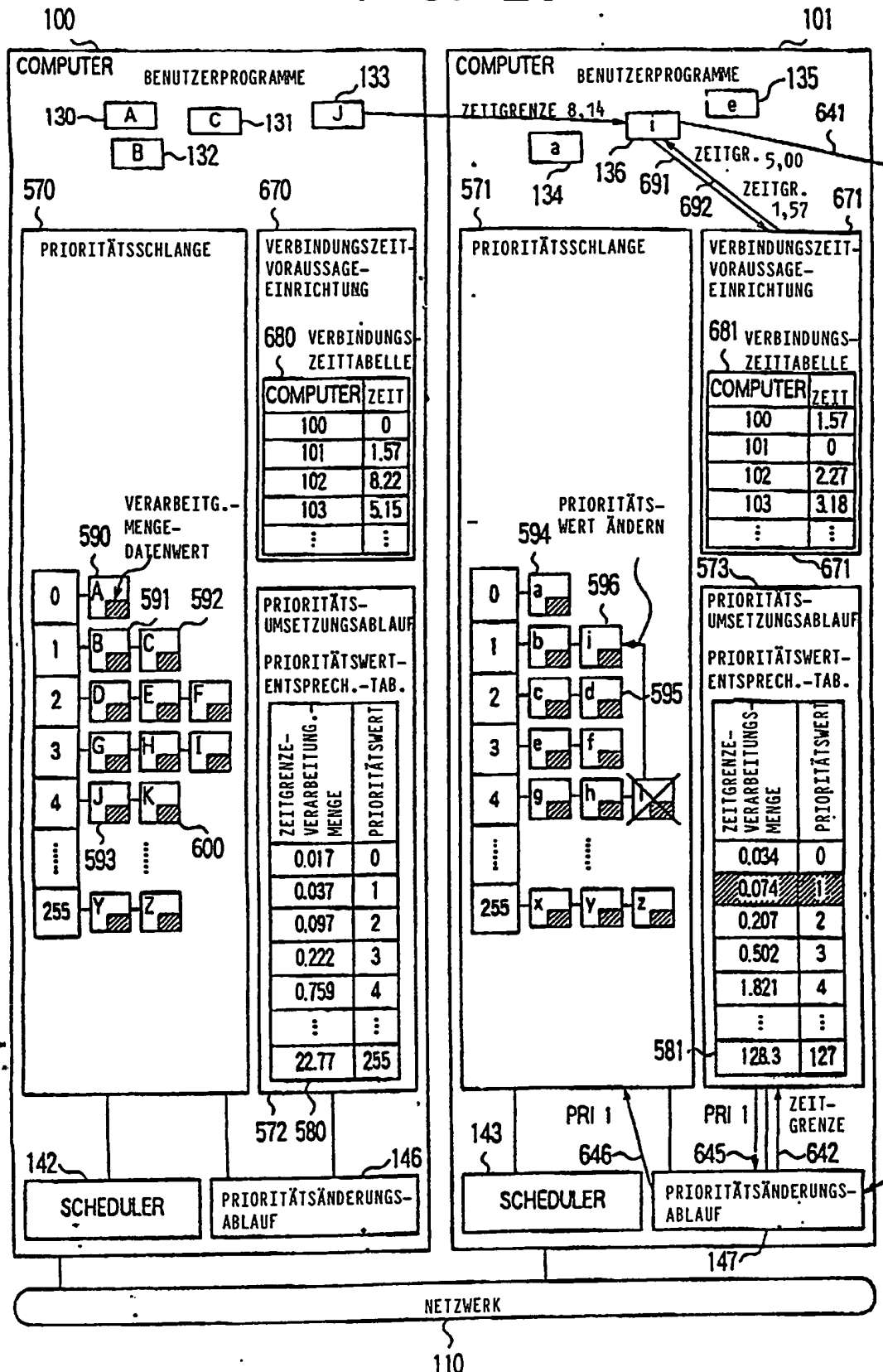


FIG. 30

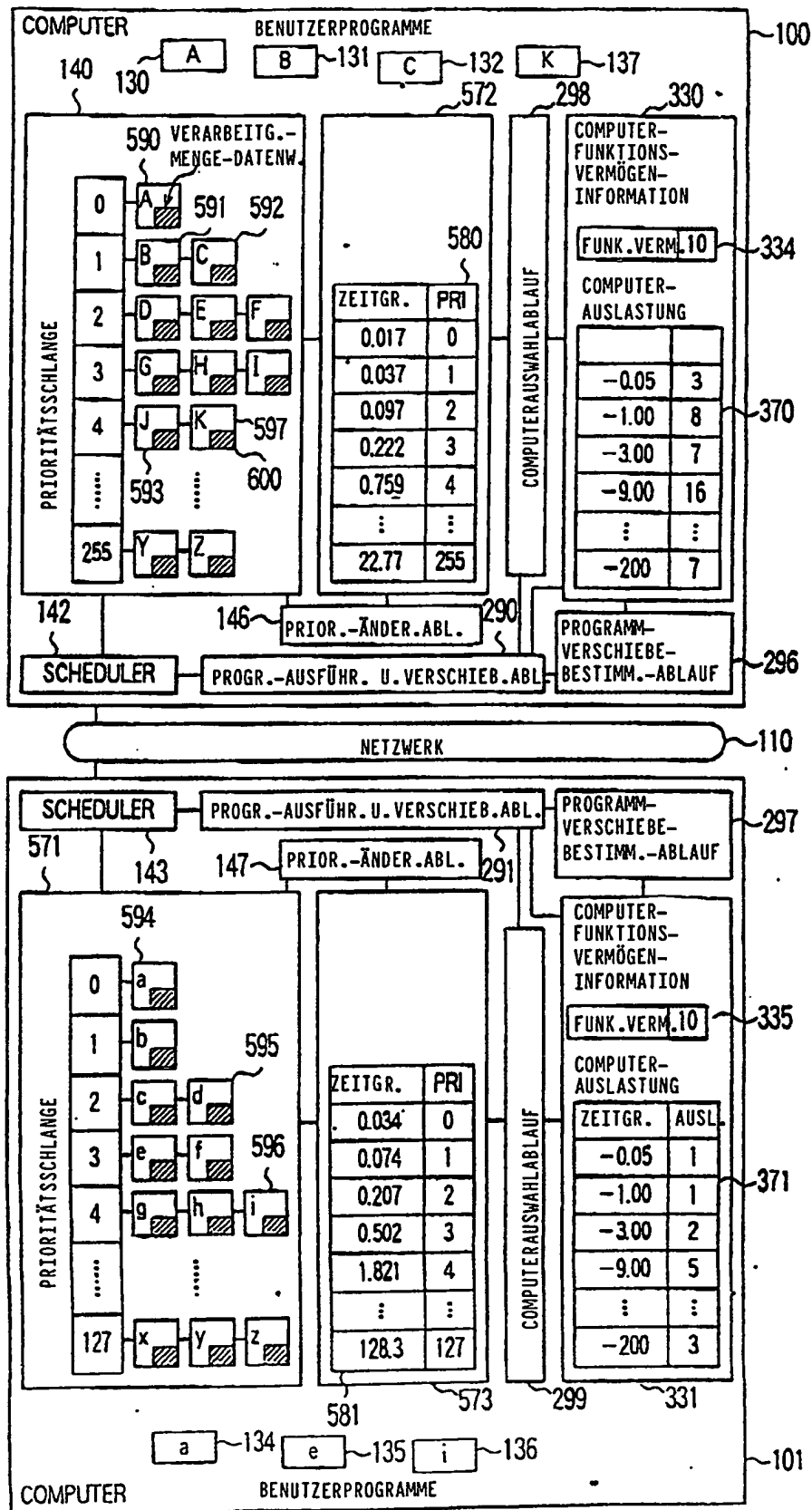
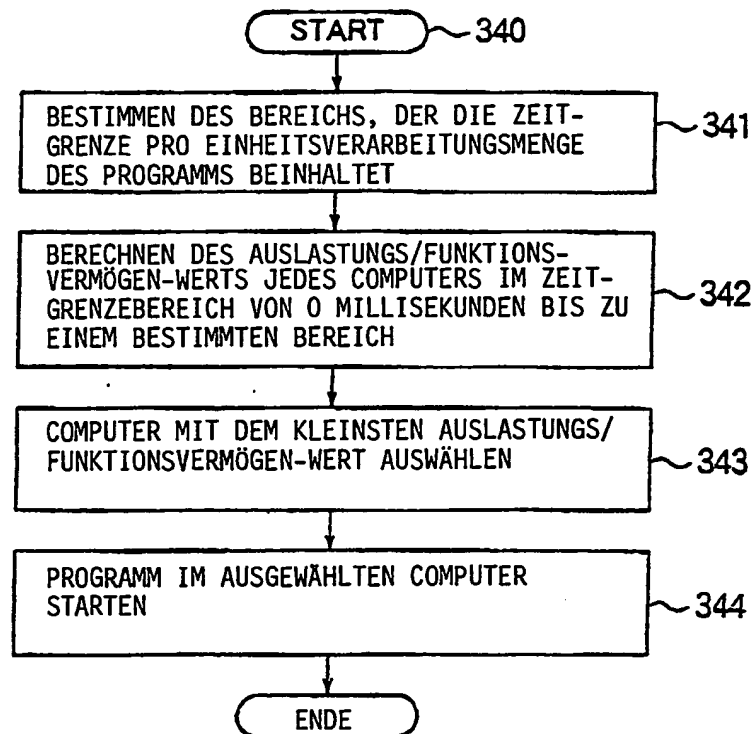


FIG. 31



## FIG. 32

## COMPUTER 100

COMPUTERAUSLASTUNG- INFORMATIONSTABELLE		AUSLASTUNGSWERT- TABELLE		AUSLASTUNGS/ FUNKTIONSVERMÖGEN- WERT-TABELLE	
ZEITGRENZE/ VERARBEIT.- MENGE	AUS- LAST.- WERT	ZEITGRENZE- BEREICH	AUS- LAST.- WERT	ZEITGRENZE- BEREICH	AUSLAST./ FUNK.-VERM.- WERT
-0.05	3	0~0.05	3	0~0.05	0.30
-1.00	8	0~1.00	11	0~1.00	1.10
-3.00	7	0~3.00	18	0~3.00	1.80
-9.00	16	0~9.00	34	0~9.00	3.40
⋮	⋮	⋮	⋮	⋮	⋮
-200	7	0~200	241	0~200	24.10

## COMPUTER 101

COMPUTERAUSLASTUNGS- INFORMATIONSTABELLE		AUSLASTUNGSWERT- TABELLE		AUSLASTUNGS/ FUNKTIONSVERMÖGEN- WERT-TABELLE	
ZEITGRENZE/ VERARBEIT.- MENGE	AUS- LAST.- WERT	ZEITGRENZE- BEREICH	AUS- LAST.- WERT	ZEITGRENZE- BEREICH	AUSLAST./ FUNKTIONS- VERM.-WERT
-0.05	1	0~0.05	1	0~0.05	0.33
-1.00	1	0~1.00	2	0~1.00	0.67
-3.00	2	0~3.00	4	0~3.00	1.33
-9.00	5	0~9.00	9	0~9.00	3.00
⋮	⋮	⋮	⋮	⋮	⋮
-200	3	0~200	71	0~200	23.67

FIG. 33

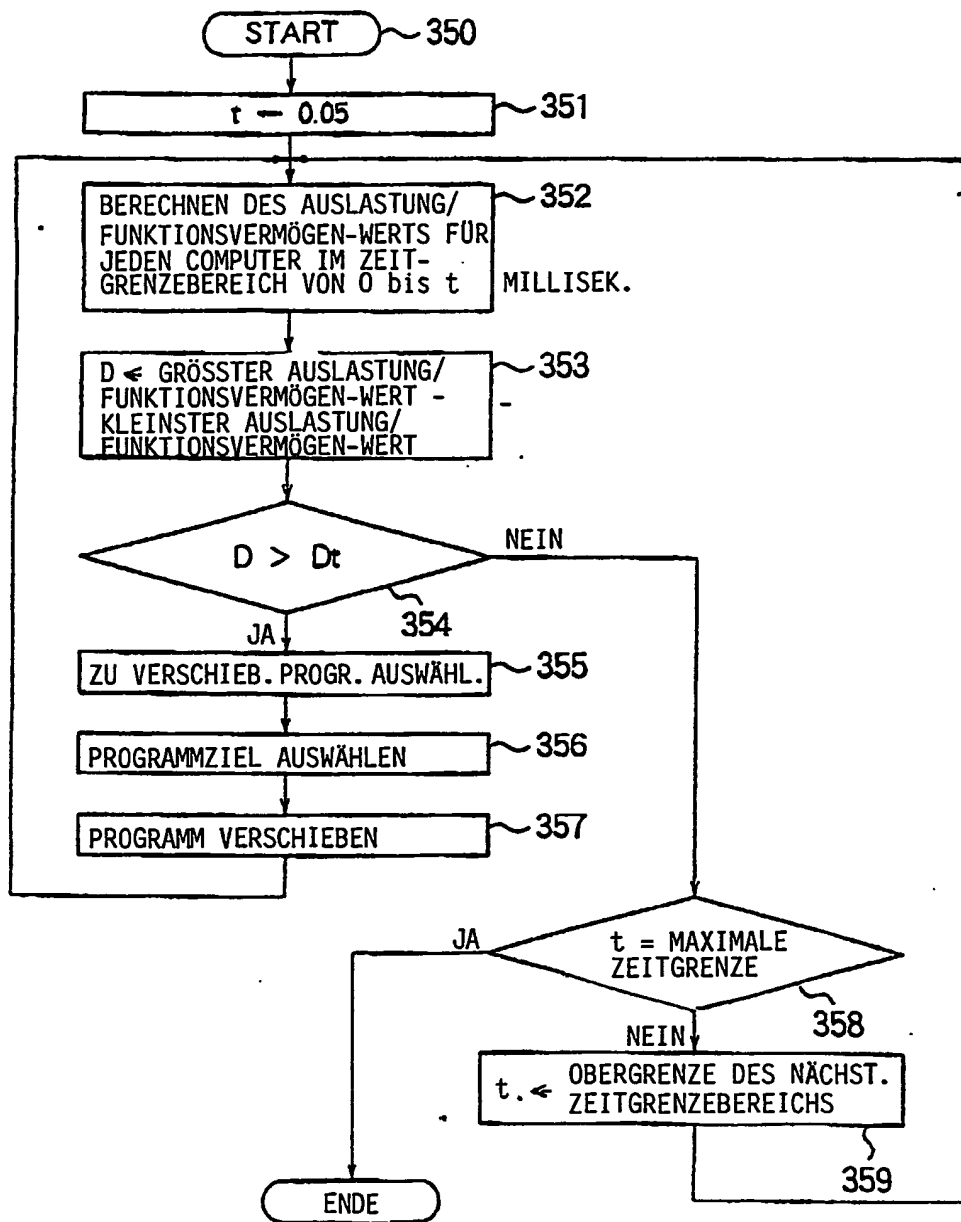


FIG. 34

COMPUTERAUSLASTUNGS- INFORMATIONSTABELLE		370	AUSLASTUNGSWERT- TABELLE		380	AUSLASTUNGS/FUNKTIONS- VERMÖGEN-WERT-TABELLE		382
ZEITGR.- VERARB.- MENGE	AUS- LAST.		ZEIT- GRENZE- BEREICH	AUS- LAST.- WERT		ZEIT- GRENZE- BEREICH	AUSL./ FUNK.- VERM.W.	
-0.05	3		0~0.05	3		0~0.05	0.30	
-1.00	7		0~1.00	10		0~1.00	1.00	
-3.00	7		0~3.00	17		0~3.00	1.70	
-9.00	16		0~9.00	33		0~9.00	3.30	
⋮	⋮		⋮	⋮		⋮	⋮	
-200	7		0~200	240		0~200	24.00	

COMPUTERAUSLASTUNGS- INFORMATIONSTABELLE		371	AUSLASTUNGSWERT- TABELLE		381	AUSLASTUNGS/FUNKTIONS- VERMÖGEN-WERT-TABELLE		383
ZEITGR.- VERARB.- MENGE	AUS- LAST.		ZEIT- GRENZE- BEREICH	AUS- LAST.- WERT		ZEIT- GRENZE- BEREICH	AUSL.- FUNK.- VERM.W.	
-0.05	1		0~0.05	1		0~0.05	0.33	
-1.00	2		0~1.00	3		0~1.00	1.00	
-3.00	2		0~3.00	5		0~3.00	1.67	
-9.00	5		0~9.00	10		0~9.00	3.33	
⋮	⋮		⋮	⋮		⋮	⋮	
-200	3		0~200	72		0~200	24.00	

FIG. 35

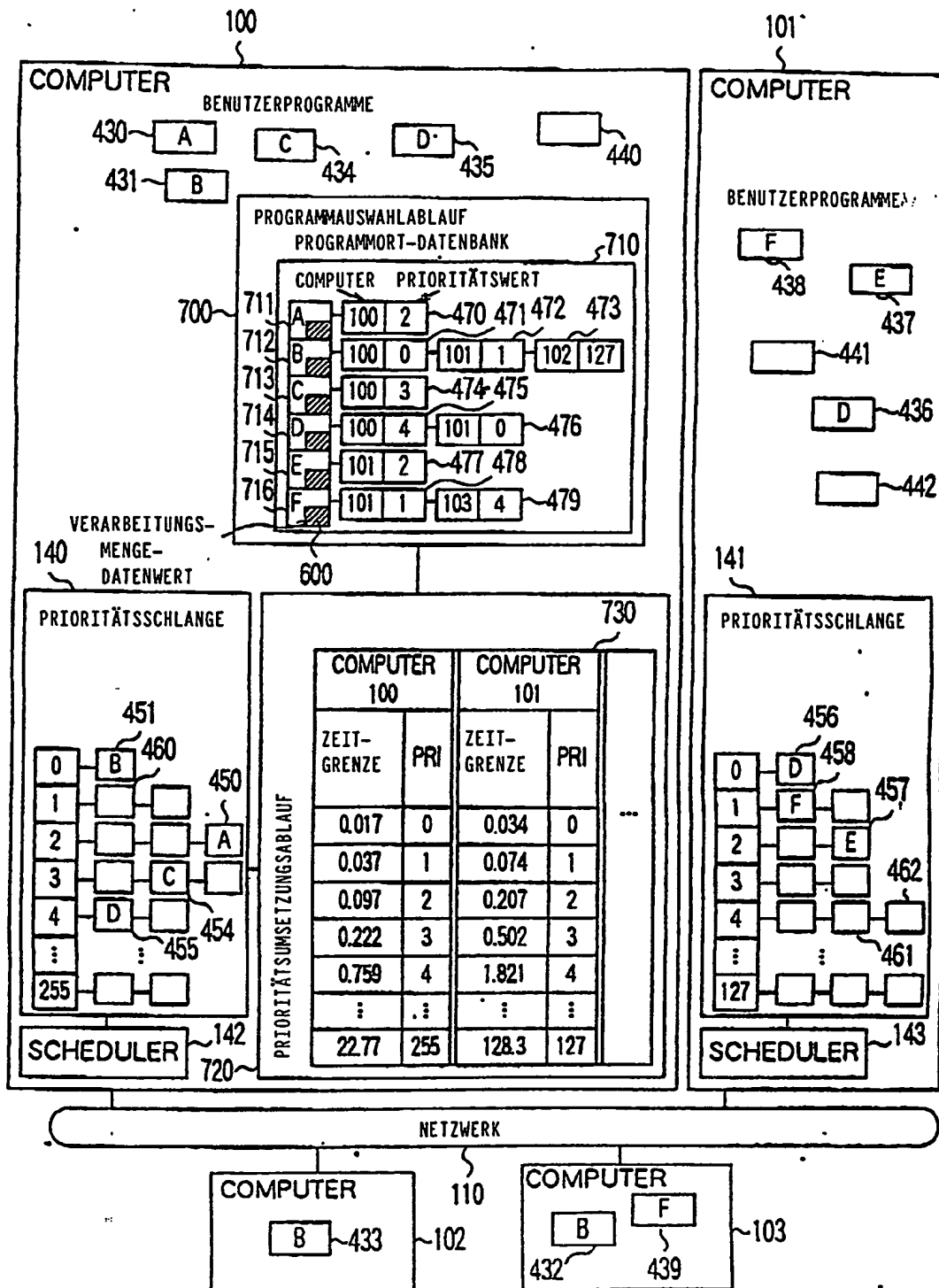




FIG. 36

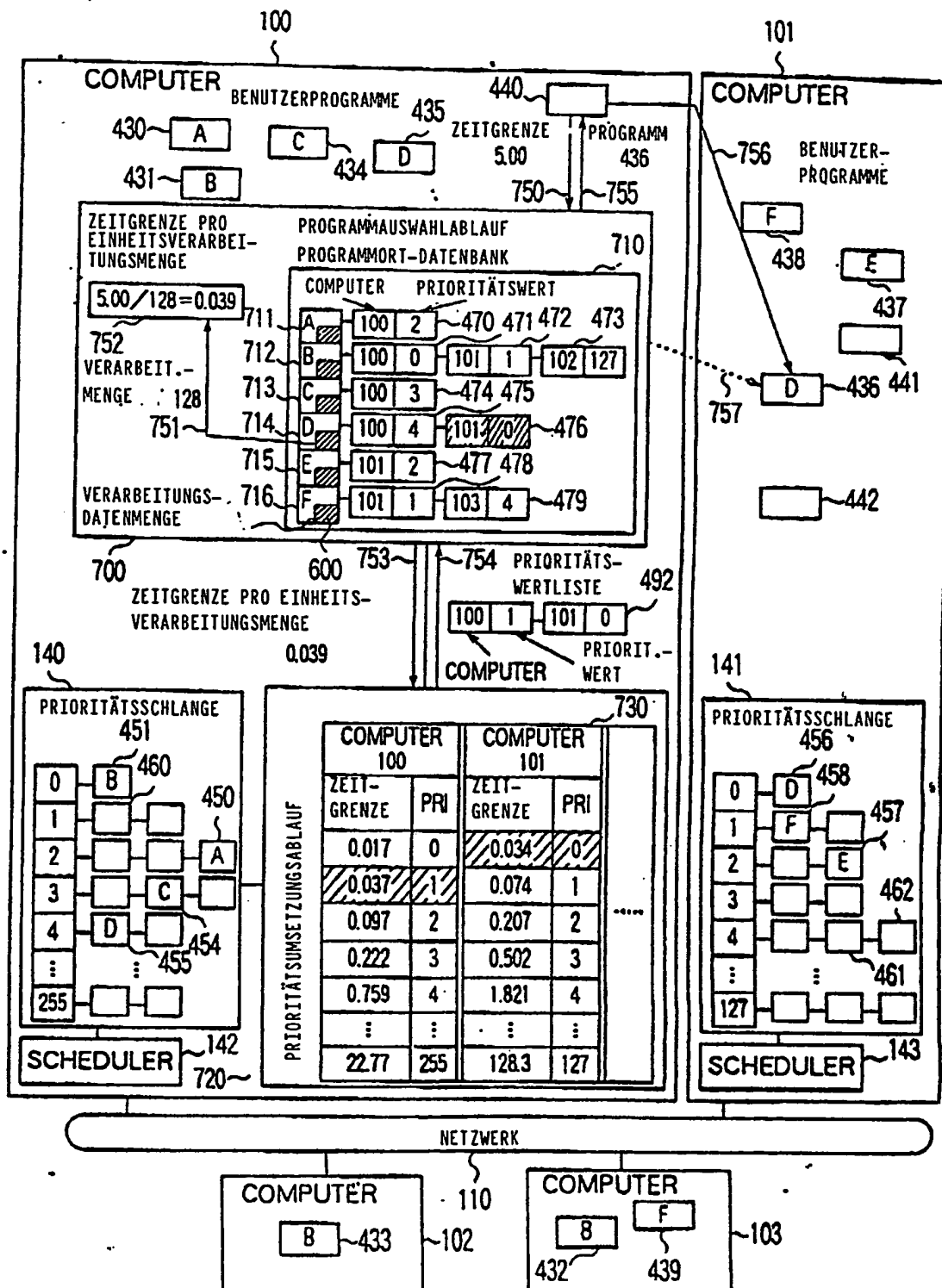


FIG. 37

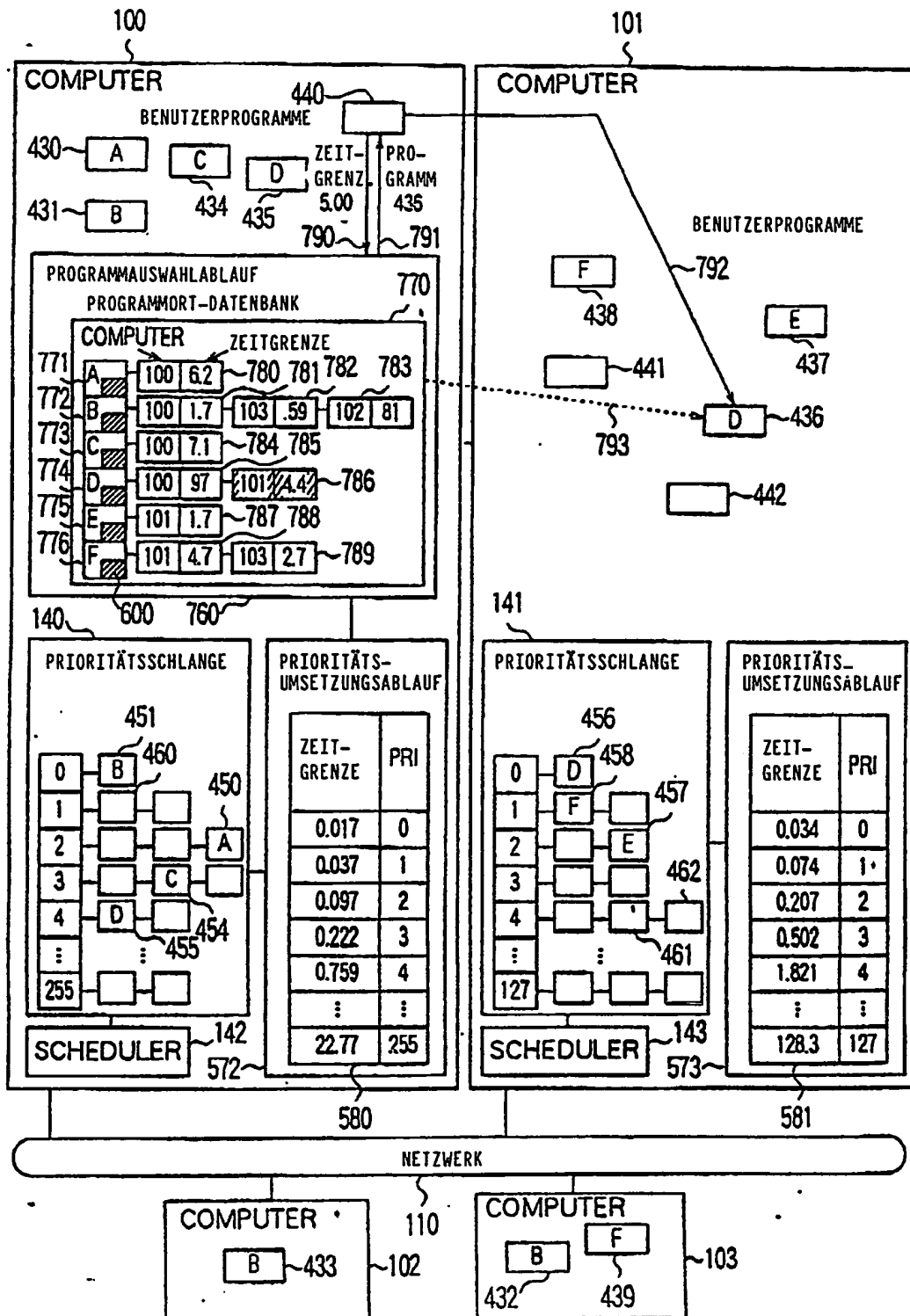


FIG. 38

